Domain embedding boundary integral equation methods and parabolic PDEs

Langdon, Stephen

Award date:
1999

Awarding institution:
University of Bath

Domain Embedding Boundary Integral Equation Methods and Parabolic PDEs

submitted by

Stephen Langdon

for the degree of Ph.D.

of the

University of Bath

1999

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Summary

In this thesis we present, analyse and implement a domain embedding boundary integral equation method for the solution of the inhomogeneous Helmholtz-type equation

$$-\Delta u + \alpha^2 u = g, \quad \text{in } \Omega \subset \mathbb{R}^2, \tag{1}$$

subject to Dirichlet or Neumann boundary conditions, where the parameter α is real and possibly large, and Ω is a bounded domain with a smooth boundary. Our method avoids altogether the explicit construction of a grid resolving Ω, and is robust as $\alpha \to \infty$.

Applications arise in the implementation of space-time boundary integral methods for reaction-diffusion equations. Discretisation in time by a linearly implicit finite difference scheme leads to a sequence of problems of the form (1), where α is proportional to $1/\sqrt{\delta t}$, δt is the time step, and the right hand side g is a function of the solutions at previous time levels.

As well as discussing the solution of (1) in detail we also use the methods which we have developed for (1) in the solution of various parabolic problems involving the heat equation and reaction-diffusion equations on general smooth bounded domains $\Omega \subset \mathbb{R}^2$.

Acknowledgements

A huge debt of thanks is owed to just about everyone I've met over the past 3+ years, but in particular I'd like to thank Ivan, without whose support and enthusiasm I wouldn't have stood a chance, my family without whom I'd have been in real trouble, all my friends and colleagues in Bath, Durham and Farnham, especially those who put up with living with me, and those who were patient with me when I burst into their offices shouting "Oh my God I'm stuck you've got to help me...", EPSRC for paying my bills for three years, Bath City FC for light relief, and last but not least Katy for putting up with me when I was spending all my time doing hard sums when all that was required was a single page stating that the answer was four point two.

Contents

Chapter 1

Introduction

This thesis is concerned with solving, numerically, boundary-value problems for the inhomogeneous Helmholtz-type equation

$$-\Delta u + \alpha^2 u = g, \quad \text{in } \Omega, \tag{1.0.1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with smooth boundary Γ, the parameter α is real and possibly large, and g satisfies certain smoothness assumptions which we will specify below.

As a first step to choosing an appropriate method for the solution of (1.0.1), we need to consider its origins. Problems of the form (1.0.1) can arise in the solution of certain parabolic partial differential equations (PDEs). For example, consider the initial boundary-value problem for the reaction-diffusion equation

$$\frac{1}{c}\frac{\partial u}{\partial t} = \Delta u + f(u), \quad \text{in } \Omega \times (0, T], \tag{1.0.2}$$

where $c, T > 0$ and f is some, possibly nonlinear, reaction term. Discretising (1.0.2) in time using a linearly implicit finite difference method (an approach known in some sections of the literature as Rothe's method) reduces it to a sequence of inhomogeneous Helmholtz problems, each of the form (1.0.1).

For example, if we choose time points $0 = t_0 < t_1 < \ldots$ then linearly implicit backward Euler timestepping gives us

$$\frac{u_{n+1} - u_n}{c\delta t_n} = \Delta u_{n+1} + f(u_n), \quad \text{in } \Omega, \quad n \geq 0, \tag{1.0.3}$$

where $\delta t_n = t_{n+1} - t_n$, and $u_n = u(\cdot, t_n)$. Rearranging, we get

$$-\Delta u_{n+1} + \frac{1}{c\delta t_n} u_{n+1} = \frac{1}{c\delta t_n} u_n + f(u_n), \quad \text{in } \Omega, \quad n \geq 0, \tag{1.0.4}$$

with corresponding boundary data. For each $n \geq 0$, (1.0.4) is of the form (1.0.1), where $\alpha = 1/\sqrt{c\delta t_n}$ is real, and where the right hand side g is given by a function of the solution at the previous time level. To solve (1.0.2) we then just need to solve (1.0.4) at each time level.

Now problems of the form (1.0.2) can occur, for example, in the pattern formation problems of mathematical biology (see eg. [10] or [64]), in which case, although the domain Ω may be complicated, it will typically have a very smooth boundary. Solving (1.0.1) on arbitrary, complicated domains using finite difference or finite element methods may become expensive, and in addition, such methods may not fully exploit the smoothness of Γ. The main aim of this thesis then is to derive and implement a method for solving (1.0.1) (and hence (1.0.2)) which avoids altogether the explicit discretisation of Ω, and makes full use of the smoothness of Γ. To this end we consider the use of boundary integral equation methods.

These methods are sometimes considered inappropriate for inhomogeneous problems, and/or problems where the solution is sought throughout a domain, because the cost of computing a particular solution to remove the inhomogeneity and/or the cost of many field point evaluations can be prohibitive. An approach which overcomes both of these difficulties within a single framework is the domain embedding method. In the first part of this thesis (Chapters 2–4), we discuss, analyse and implement this method for the solution of (1.0.1). Our method avoids altogether the explicit discretisation of Ω, and is robust as $\alpha \to \infty$ (or equivalently as $\delta t_n \to 0$ in (1.0.4)).

We give a more detailed introduction to this in §1.1, as well as presenting some background theory on boundary integral equation methods, and describing in more detail those parts of our method for solving (1.0.1) which are original.

In the second part of this thesis (Chapters 5 and 6) we then consider the application of boundary integral equation methods to the solution of parabolic problems such as (1.0.2). We give a more detailed introduction to this in §1.2.

1.1 The Inhomogeneous Helmholtz Problem

The natural method for solving homogeneous elliptic PDEs such as Helmholtz's equation, without discretising the domain, is the boundary integral equation method. This is based on Green's Theorem, see for example [16, Theorem 2.1]. In this theorem, and throughout the rest of this thesis (except where explicitly stated), we shall assume that $\Omega \subset \mathbb{R}^2$ is a bounded domain whose boundary Γ is parametrised by a 2π-periodic function $\gamma : \mathbb{R} \mapsto \Gamma$ which satisfies

$$\gamma \in \mathcal{C}^\infty(\mathbb{R}), \tag{1.1.5}$$

with

$$\gamma(s) = \gamma(t) \text{ if and only if } s - t \in 2\pi i, \quad i \in \mathbb{Z}, \tag{1.1.6}$$

and

$$|\gamma'(t)| > 0, \quad \text{for all } t \in \mathbb{R}. \tag{1.1.7}$$

Of course many of the statements which we shall make will hold for much more general regions (e.g. with corners) but since we are concerned here with a different type of novelty (as described above) we avoid technical detail arising from nonsmooth Γ.

Theorem 1.1.1 *Let $u \in \mathcal{C}^2(\bar{\Omega})$, and let $\partial/\partial n$ denote differentiation with respect to the unit outward normal vector to Γ. Then we have* Green's formula *(also known in the literature as the* Helmholtz representation*)*

$$\begin{aligned} u(\boldsymbol{x}) \;=\; & \int_\Gamma \left\{ \frac{\partial u}{\partial n}(\boldsymbol{y}) \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) - u(\boldsymbol{y}) \frac{\partial \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})}{\partial n(\boldsymbol{y})} \right\} ds(\boldsymbol{y}) \\ & + \int_\Omega \left\{ -\Delta u(\boldsymbol{y}) + \alpha^2 u(\boldsymbol{y}) \right\} \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y}, \quad \boldsymbol{x} \in \Omega, \end{aligned} \tag{1.1.8}$$

where Φ_α (see (2.1.5)) denotes the fundamental solution of the two-dimensional Helmholtz equation

$$-\Delta u + \alpha^2 u = 0, \quad \text{in } \Omega. \tag{1.1.9}$$

Proof: For a full proof see [16, p.17]. □

Remark: Similar representations apply to more general elliptic problems, provided a fundamental solution is known.

From (1.1.8) it is clear that if u were a solution of the homogeneous Helmholtz equation (1.1.9) then we would have the formula

$$u(\boldsymbol{x}) = \int_\Gamma \left\{ \frac{\partial u}{\partial n}(\boldsymbol{y}) \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) - u(\boldsymbol{y}) \frac{\partial \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})}{\partial n(\boldsymbol{y})} \right\} ds(\boldsymbol{y}), \quad \boldsymbol{x} \in \Omega, \qquad (1.1.10)$$

which represents the solution u in terms of the boundary data $u|_\Gamma$ and $\partial u/\partial n|_\Gamma$.

The classical approach to solving the homogeneous problem (1.1.9) using boundary integral equation methods is to take the equation (1.1.10) and let the point $\boldsymbol{x}$ approach the boundary. Using the classical *jump relations* of potential theory (see [15] or [49]) we then get the integral equation

$$\frac{1}{2} u(\boldsymbol{x}) = \int_\Gamma \left\{ \frac{\partial u}{\partial n}(\boldsymbol{y}) \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) - u(\boldsymbol{y}) \frac{\partial \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})}{\partial n(\boldsymbol{y})} \right\} ds(\boldsymbol{y}), \quad \boldsymbol{x} \in \Gamma, \qquad (1.1.11)$$

which involves values of u and its normal derivative only on the boundary. Hence if either the Dirichlet or the Neumann boundary data is known, (1.1.11) is an integral equation for the unknown complementary boundary data, which can be solved (see Chapter 3), and for points $\boldsymbol{x} \in \Omega$ the value of u can then be found by using the representation (1.1.10).

In this case the method would only involve boundary integrals. Hence there would be no need to discretise the domain, and in Chapters 3 and 4 we describe in more detail how we can solve the homogeneous problem (1.1.9) quickly. In Chapter 3 we show how to formulate and solve (1.1.11), and in Chapter 4 we discuss the cost of the method and we show how we can evaluate u quickly at many points throughout the domain.

However, here we are also concerned with the problem of solving (1.0.1) when $g \not\equiv 0$, and it is clear that in this case using the integral equation (1.1.8) would require the evaluation of domain integrals over the whole of Ω. Direct numerical integration of the domain integrals by quadrature formulae is in some sense a "classical" approach, and an error analysis can be performed to justify this method mathematically. The problem with this approach is that we still have

in some sense to devise a mesh on Ω, which somehow defeats the purpose of the boundary integral equation method. Moreover, evaluating a domain integral can be expensive, especially if Ω is complicated.

Another way to solve the boundary value problem for (1.0.1) is to find a particular solution u_p which satisfies (1.0.1), but does not necessarily satisfy any given boundary conditions. If such a function can be found, then by linearity, the function $v = u - u_p$ satisfies the homogeneous Helmholtz equation (1.1.9), with boundary conditions derived from those of u and u_p. The integral formulation for v is then free of domain integrals, and v can then be found using the methods of Chapters 3 and 4.

This is the approach we shall use, and it can be considered as a three step algorithm. For example, for the problem (1.0.1), supplemented by Dirichlet data $u = \phi$ on Γ, the three steps are as follows.

i). Find a particular solution u_p which satisfies $-\Delta u_p + \alpha^2 u_p = g$ in Ω.

ii). Solve the homogeneous problem $-\Delta v + \alpha^2 v = 0$ in Ω, with boundary data $v = \phi - u_p$ on Γ, by solving the boundary integral equation (1.1.11), to find the complementary boundary data $\partial v/\partial n$ on Γ.

iii). Evaluate v quickly throughout Ω, using the values of $\partial v/\partial n$ and $v|_\Gamma$, and (1.1.10).

The solution of the Dirichlet problem for (1.0.1) is then given by $u = v + u_p$. The Neumann problem can be solved in a similar way.

An approach which allows the implementation of steps (i)–(iii) in a single framework is the domain embedding method, originally developed by Mayo and co-workers [58]–[62]. The basic tool in this method is to embed the domain Ω in a fictitious domain (rectangular in two dimensions) on which finite differences fit nicely.

Even working within this single framework, the three components of the method are still quite distinct. Hence we devote a separate chapter to each. A full introduction and literature review for each component can be found separately within each chapter, but below we briefly summarise some of the main ideas involved.

1.1.1 Finding a particular solution

In Chapter 2 we consider the problem of finding a particular solution of (1.0.1). We do this by solving an extended problem on the fictitious domain, and this can be done quickly using finite differences and fast Fourier transforms. Our method is based on that in [60]. The main new results which we have obtained here are:-

- The application of the method to the Helmholtz equation, rather than Poisson's equation.
- Theoretical justification for the use of the method by means of a result regarding the continuity of a certain particular solution of (1.0.1). For clarity this result is presented in Appendix C, but it is used in Chapter 2.
- Some new ideas for the practical implementation of the method (see especially §2.5).

1.1.2 Solving the homogeneous Helmholtz equation

In Chapter 3 we consider the problem of solving the first and second kind boundary integral equations arising from (1.1.9). Standard methods for solving these integral equations (see for example Kress [49, Chapter 12]) may break down as α becomes large. In the application to (1.0.4), α may be very large if the time step is small. Hence we have developed a new collocation method for solving the integral equations which has a robust convergence rate as $\alpha \to \infty$. All of the work in this chapter is original, and the main new results which we have obtained here are:-

- Numerical results demonstrating how standard methods may break down as α becomes large.
- Description of a new collocation method for solving both the first and second kind boundary integral equations arising from (1.1.9).
- A complete error analysis for the fully discrete collocation method applied to the second kind boundary integral equations, in which we prove that the convergence rate of the method is independent of α.

- Numerical results demonstrating the robustness of our new method with respect to α for both the first and second kind boundary integral equations.

We also describe and analyse the method as applied to the second kind boundary integral equations in [53].

1.1.3 Evaluating the solution throughout Ω

In Chapter 4 we consider the cost of computing the solution of (1.1.9) at many points in Ω. This could be done by using the formula (1.1.10) directly, but this may be expensive. Instead, we use a method based on that in Mayo [59]. The basic idea is to evaluate the solution directly at only a small number of carefully chosen points near Γ, and then to use a fast solver to extend the solution to the rest of the domain. However, the numerical evaluation of the integrals on the right hand side of (1.1.10) may be difficult if $\boldsymbol{x}$ is near Γ. Hence we have developed a new method for evaluating the solution of (1.1.9) when $\boldsymbol{x}$ is near Γ which completely avoids the use of (1.1.10). The main new results which we have obtained in this chapter are:-

- The application of the method of [59] to Helmholtz's equation, rather than Laplace's equation.
- Some new ideas for the numerical evaluation of the integrals on the right hand side of (1.1.10).
- Description and implementation of a new method for evaluating the solution of (1.1.9) near Γ which avoids the use of the formula (1.1.10) altogether.

1.2 Solving Parabolic PDEs

Having developed a method for solving the inhomogeneous Helmholtz-type equation (1.0.1), we then want to apply it to the solution of parabolic PDEs of the form (1.0.2).

We begin in Chapter 5 by considering integral equation methods for the heat equation ((1.0.2) with $f \equiv 0$). Solving the heat equation by Rothe's method has been dealt with in the literature (see for example Chapko and Kress [12],

Lubich and Schneider [56]), but existing integral equation methods for solving the spatial equations may break down when the time step becomes small. We demonstrate in Chapter 5 how combining the collocation method of Chapter 3 with the approach in [12] can lead to improved accuracy as the time step becomes small, and we also present a new error analysis for this combined method. We also discuss this idea in [52].

In Chapter 6 we then discuss the application of the domain embedding method of Chapters 2–4 to the solution of nonlinear reaction-diffusion equations of the form (1.0.2), and we present numerical results demonstrating the good performance of the method when applied to Fisher's equation ((1.0.2) with $f(u) = u(1-u)$).

Substantial portions of Chapters 5 and 6 are also original, although one purpose of Chapter 5 is also to describe diverse existing results on integral equation methods for parabolic problems in a unified way.

1.3 Notation

Throughout this thesis, the notation we use for the various function sets and spaces in which we work is fairly standard, but for clarity it is summarised in Appendix A.

Note that certain parameters (n, N, m etc.) may take on different meanings in different chapters. Where this is the case the specific use of each parameter within the chapter will be discussed in the introduction to that chapter.

Chapter 2

Finding a particular solution

2.1 Introduction

In this chapter, we will be concerned with the problem of finding a particular solution of (1.0.1), which we repeat here for clarity,

$$-\Delta u + \alpha^2 u = g, \quad \text{in } \Omega \subset \mathbb{R}^2. \tag{2.1.1}$$

As described in Chapter 1, the situation we are interested in is when Rothe's method is applied to the solution of parabolic PDEs, in which case the unknown function u in (2.1.1) represents the solution of a parabolic PDE at some given time level, and the right hand side g is a function of the solutions at previous time levels. Hence the right hand side g is not in general known as an explicit function, but rather it is given as data. This must be considered when seeking a particular solution of (2.1.1).

We begin by defining what is meant by the *fundamental solution* (see for example [20] or [41]) of a linear elliptic PDE of the form $Lu = 0$, such as the Helmholtz equation

$$\Delta u + k^2 u = 0, \quad \text{in } \Omega. \tag{2.1.2}$$

Definition 2.1.1 *The* fundamental solution *of a linear elliptic PDE, $Lu = 0$ in Ω, is a function $\Phi(\boldsymbol{x}, \boldsymbol{y})$ defined for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$, $\boldsymbol{x} \neq \boldsymbol{y}$, which for every fixed*

$\boldsymbol{y} \in \Omega$ satisfies as a function of $\boldsymbol{x}$

$$L_x \Phi(\boldsymbol{x}, \boldsymbol{y}) = \delta(\boldsymbol{x} - \boldsymbol{y}),$$

where L_x denotes the operator L where differentiation is with respect to $\boldsymbol{x}$, and δ is the Dirac delta function. Equivalently

$$\int_\Omega L_x \{\Phi(\boldsymbol{x}, \boldsymbol{y})\} f(\boldsymbol{y}) d\boldsymbol{y} = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{2.1.3}$$

for every continuous function $f : \Omega \mapsto \mathbb{R}$.

It can be shown (see for example [14, pp.339-341]) that a fundamental solution of the two-dimensional Helmholtz equation (2.1.2) is given by

$$\Phi(\boldsymbol{x}, \boldsymbol{y}) := \frac{i}{4} H_0^{(1)}(k|\boldsymbol{x} - \boldsymbol{y}|), \quad \boldsymbol{x} \neq \boldsymbol{y}, \tag{2.1.4}$$

where $H_0^{(1)}$ is the first kind Hankel function of order zero. In the case we are particularly interested in, where $k = i\alpha$, $\alpha \in \mathbb{R}$, $\Phi(\boldsymbol{x}, \boldsymbol{y})$ is equal to

$$\Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) := \frac{1}{2\pi} K_0(\alpha|\boldsymbol{x} - \boldsymbol{y}|), \quad \boldsymbol{x} \neq \boldsymbol{y}, \tag{2.1.5}$$

where K_0 is the modified Bessel function of order zero. For a detailed introduction to Hankel functions and modified Bessel functions we refer to [1], but for clarity we have summarised some of their most important properties in Appendix B.

Using (2.1.3) it is clear that one way of constructing a particular solution of (2.1.1) is to compute the domain integral

$$U(\boldsymbol{x}) := \int_\Omega \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{y}) d\boldsymbol{y}. \tag{2.1.6}$$

However, there are two difficulties with using this approach. The kernel Φ_α of the integral (2.1.6) is logarithmically singular at $\boldsymbol{x} = \boldsymbol{y}$, and this means that standard quadrature schemes would be inaccurate.

Even under the assumption that one has an efficient way of dealing with the logarithmic singularity, direct application of some quadrature formula to evaluate (2.1.6) can be very expensive, as explained in Mayo [60]. For example, to

compute (2.1.6) at every point of an $n \times n$ grid using a quadrature scheme such as that of Atkinson [5] would have an operation cost of $\mathcal{O}(n^4)$.

Fortunately, several ingenious methods of getting around these problems have recently been proposed in the literature. In §2.2 we present a literature review in which we briefly describe some of the methods which have been suggested for finding a particular solution of (2.1.1). We also explain why we have chosen to use the domain embedding method.

The basic idea of this method is to extend the problem onto a rectangle, cover the rectangle with a uniform mesh, and then solve the extended problem on this mesh. Because of the uniformity of the mesh, fast methods can be used for the latter task. The main difficulty comes in extending the problem from the irregular domain to the rectangle in such a way that the solution of the extended problem is a sufficiently accurate approximation to a particular solution of (2.1.1). In order to do this one needs to compute special correction terms at certain "irregular" mesh points (see Definition 2.3.2) close to the boundary of Ω.

In §2.3 we describe precisely how to construct the extended problem on the rectangle. In §2.4 we then describe how the extended problem can be solved quickly using finite differences and fast Fourier transforms. In §2.5 we describe the hidden cost of the method, namely that of computing the irregular mesh points, and we present some new computational algorithms for doing this. Finally, in §2.6 we apply the method to some simple examples and present some numerical results.

2.2 Literature review

When seeking a particular solution of (2.1.1), the first thing to note is that if the inhomogeneous term g were simple, for example a polynomial or a series of trigonometric functions, then a particular solution could be found very easily by using the method of undetermined coefficients (see for example [5]). In general this will not be possible, but if g could be approximated reasonably accurately by some function for which (2.1.1) can be solved easily, then this might provide a good method for finding a particular solution. This is the idea used in much of the recent literature. The methods for finding a particular solution then fall into two basic classes.

i). Approximating the inhomogeneous term g in such a way that a particular solution of (2.1.1) can be found more easily.

ii). Direct numerical approximation of the domain integral (2.1.6). Much work has been done on speeding up the domain integration so that the cost is not excessive.

There is some overlap between these two approaches, notably in the work of Golberg [26], but for clarity we treat them individually. We also summarise the reasons why the domain embedding approach that we actually use, which is based on the method of Mayo [60], is superior to the other methods for our particular needs.

2.2.1 Approximating the inhomogeneous term

The most widely used method in the literature for evaluating particular solutions seems to be the Dual Reciprocity Method, developed by Brebbia et al. [66]. This consists of approximating the inhomogeneous term g in (2.1.1) by a linear combination of globally defined basis functions $\{\phi_j\}$, $j = 1, \ldots, N$, i.e.

$$g \approx \tilde{g} := \sum_{j=1}^{N} \lambda_j \phi_j. \tag{2.2.7}$$

The basis functions $\{\phi_j\}$ are chosen in such a way that the equations

$$-\Delta \psi_j + \alpha^2 \psi_j = \phi_j, \quad j = 1, \ldots, N, \tag{2.2.8}$$

can be solved explicitly, or at least more easily than the original problem. The approximate particular solution defined by

$$\tilde{u}_p := \sum_{j=1}^{N} \lambda_j \psi_j$$

will then satisfy

$$-\Delta \tilde{u}_p + \alpha^2 \tilde{u}_p = \sum_{j=1}^{N} \lambda_j \phi_j = \tilde{g}.$$

The question now is how to choose the basis functions ϕ_j. These have to be chosen in such a way that they satisfy the following two properties.

i). The equations (2.2.8) must be easy to solve.

ii). The approximation of g in (2.2.7) must be sufficiently accurate.

A simple choice for the basis functions would be either polynomials or trigonometric polynomials. These would certainly satisfy property (i). However, for an arbitrary domain Ω they are unsuitable, because of the general unavailability of methods for computing good approximations using such functions. For example, finding good interpolation points on a general two-dimensional region is not always obvious.

Most of the recent literature concentrates on the use of various radial basis functions (see for example [24], [25], [27]). There exists an extensive approximation theory for these functions (see for example [68], [69]), but there seems to be little agreement on which are the best ones to use.

The approach of approximating g in order to simplify (2.1.1) has a vast literature, and the method has been applied successfully to many problems, with good numerical results being reported. Much of the pre-1990 work is treated in detail in the book by Partridge, Brebbia and Wrobel [66], and a lot of the post-1990 work is summarised by Golberg [25], where there is also a definitive bibliography on the subject.

A major problem however is that there is very little mathematical justification for the method in terms of an error analysis, and the questions of convergence and stability have not been fully addressed. Much work has been done on obtaining error bounds for the numerical approximation of g, but as we are seeking a non-unique particular solution, rather than the solution of a boundary value problem, the usual error analysis for second order linear elliptic boundary value problems cannot be applied directly, and so this alone does not guarantee the accuracy of the numerical approximation to the particular solution.

2.2.2 Numerical integration

The basic idea of this approach is to evaluate the domain integral (2.1.6) directly using some numerical integration method. The main advantage is that because

the method is based on standard numerical integration techniques, an error analysis can be performed. Various approaches can be used to avoid direct domain discretisation, and to deal with the singularity in the kernel.

The most widely known idea seems to be Atkinson's method [5]. This requires that one can extend the inhomogeneous function g smoothly to a suitably simple domain $\hat{\Omega} \supset \bar{\Omega}$, where $\bar{\Omega}$ is the closure of Ω, and then defines

$$\tilde{u}_p(\boldsymbol{x}) := \int_{\hat{\Omega}} \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{y}) d\boldsymbol{y}, \quad \boldsymbol{x} \in \bar{\Omega}. \tag{2.2.9}$$

Then $\tilde{u}_p$ is also a particular solution, although it is not equal to the particular solution U given by (2.1.6). Hence this approach could not be used if one wanted to use the formulation (1.1.8) directly, although a simple modification of (1.1.8) could be used. By choosing $\hat{\Omega}$ wisely (see below), $\tilde{u}_p$ can be evaluated a lot more easily than U.

In [5], Atkinson chooses $\hat{\Omega}$ to be an ellipse, and then makes a simple substitution to allow the integral to be evaluated using one-dimensional quadrature formulae (Gaussian quadrature and the trapezoidal rule). Golberg and Chen have extended this method (see [26]), by using a radial basis function approximation for g in (2.2.9) so that some of the integrals arising after the substitution has been made can be evaluated analytically. This method gives faster convergence than in Atkinson's original paper [5], but despite some theory in [27] it still seems to lack formal mathematical justification. In [26], the question of how to optimally choose the lengths of the semiaxes of the ellipse $\hat{\Omega}$ is also addressed.

Although Atkinson's method seems an improvement over direct numerical integration by domain discretisation, it has a major drawback which renders it unusable for our purposes. The method will work well when the right hand side g of (2.1.1) is a known function that can be smoothly extended onto the ellipse. However, if g is not known explicitly then it is not clear how to extend it smoothly. In [2] it is shown that in theory it is always possible to extend a suitably well-behaved function smoothly from one domain with a smooth boundary to another, but it is not clear from [2] or [5] how to *actually compute* a smooth extension onto a larger domain when the function g is known only as data (i.e. point values of g can be computed, but a *formula* for g is not known). This will be the case in general for our problem.

Another difficulty with methods based on domain integration is that the domain integral must be evaluated at each point where the solution is required, and hence if the particular solution is required throughout the domain then the method may become very expensive. As mentioned in §2.1, the operation cost for computing (2.2.9) at every point of an $n \times n$ grid using Atkinson's method would be $\mathcal{O}(n^4)$.

Other methods have also been suggested for speeding up the numerical integration. The multiple reciprocity method (see for example [66, p.45]) can be used to convert the domain integral (2.1.6) into an infinite sum of line integrals. However, it requires g to be a known function whose derivatives can be evaluated analytically, and so as explained above it is unusable for our purposes.

Another idea is the use of Monte-Carlo or Quasi-Monte-Carlo integration on an extended rectangular domain (see for example [13]). However, again the question of how to extend the inhomogeneous term smoothly is not fully addressed, and the rate of convergence seems to be a lot slower than that for the domain embedding method which we will actually use.

2.2.3 Mayo's domain embedding approach

The method we will actually employ is a domain embedding method, based on that developed by Mayo in [60]. This can be thought of as a fast method for evaluating the domain integral (2.1.6) at many points, but it does not employ any numerical integration. The key to the method is not to evaluate the integrals directly but rather to solve the underlying differential equation on an extended domain.

As described in §2.1, the basic idea is to embed the domain Ω in a fictitious domain (rectangular in two dimensions) on which finite differences fit nicely, and then to discretise the fictitious domain using a uniform mesh. Solving an extended problem on the rectangular domain gives a particular solution to the inhomogeneous problem. As we shall show in §2.3–2.4, this particular solution can be found to $\mathcal{O}(h^2)$ accuracy, with a cost of $\mathcal{O}(n^2 \log n)$, using finite differences and fast Fourier transforms, where h represents the mesh diameter and n^2 is the number of mesh points. The method is superior, for our needs, to the methods outlined in §2.2.1–2.2.2 for the following reasons.

- There is no need to compute a continuous extension of g onto the extended domain.
- In the context of the whole problem we are trying to solve, namely (2.1.1) with given Neumann or Dirichlet boundary data, the domain embedding approach gives us a single framework to work in. This is because it also gives us a way of computing the solution of the boundary value problem for (2.1.1) at many points in Ω, without having to construct a mesh on Ω (see also [62] and Chapter 4).
- The method is in some sense optimal ($\mathcal{O}(h^2)$ accuracy, $\mathcal{O}(n^2 \log n)$ cost).
- There is a better developed theory for the method than for the methods of §2.2.1.
- Unlike the numerical integration methods of §2.2.2, the particular solution can be computed quickly throughout the domain Ω.

To date, the approach in [60] has only been developed for the Laplace operator. One of the aims of this thesis is to extend and test it in the context of the Helmholtz-type equation (2.1.1).

Before describing Mayo's method in greater detail, we mention at this point that algorithms are presented in Vainikko [82] for solving second kind integral equations with integral operators such as (2.1.6). Vainikko's method has similarities to Mayo's method in that a uniform mesh is used and the domain is embedded in a larger uniform rectangular mesh so that, making use also of the convolution nature of the integral operator, the 2D FFT can be used. This method attains $\mathcal{O}(h^2 \log^2 h)$ accuracy with a cost of $\mathcal{O}(n^2 \log n)$.

However, we emphasise that Vainikko's method is for solving integral equations rather than for computing domain integrals, and so as it stands it is not really comparable with Mayo's method. It may be interesting in the future to attempt to adapt Vainikko's method to solve our problem, and then compare its performance with Mayo's method.

2.3 Constructing the extended problem

In this section we show how our domain embedding method leads to a problem on an extended rectangular domain $R \supset \bar{\Omega}$, the solution of which is a second order accurate approximation to a particular solution of (2.1.1) and can be evaluated at many points in Ω without the need to compute any domain integrals directly. The question of how to actually solve the extended problem efficiently is addressed in §2.4.

As a first step we embed the domain Ω in a larger domain R on which finite differences fit nicely. For simplicity, we take R to be the square of side length L centred at the origin,

$$R := \left\{(x,y) : -\frac{L}{2} < x < \frac{L}{2},\ -\frac{L}{2} < y < \frac{L}{2}\right\}. \tag{2.3.10}$$

For certain very elongated domains it might be preferable to use a rectangle with different height and width, but we do not discuss this generalisation here. The next step is to cover R in a uniform mesh. Again for simplicity we take the same number of mesh points n in each direction, and the mesh points are then given by

$$(x_i, y_j) = \left(-\frac{L}{2} + \frac{iL}{n+1}, \frac{L}{2} - \frac{jL}{n+1}\right), \quad i,j = 0,\ldots,n+1, \tag{2.3.11}$$

and the mesh width is given by

$$h = \frac{L}{n+1}. \tag{2.3.12}$$

We denote the set of interior mesh points by

$$R_h := \{(x_i, y_j):\ i,j = 1,\ldots,n\}, \tag{2.3.13}$$

and the set of boundary mesh points by

$$\begin{aligned} \partial R_h \ :=\ & \{(x_i, y_j):\ i = 0 \text{ or } n+1,\ j = 0,\ldots,n+1\} \\ & \cup \{(x_i, y_j):\ i = 0,\ldots,n+1,\ j = 0 \text{ or } n+1\}. \end{aligned}$$

Remark: Because of the way in which we will solve the extended problem on the square, using Fast Fourier Transforms (see §2.4), in practice we will always choose n such that

$$n = 2^k - 1, \quad k = 1, 2, \ldots .$$

We now make a few definitions.

Definition 2.3.1 *For a given mesh point (x_i, y_j), its four* nearest neighbours *are defined to be the points (x_{i+1}, y_j), (x_{i-1}, y_j), (x_i, y_{j+1}) and (x_i, y_{j-1}).*

Definition 2.3.2 *A mesh point $(x_i, y_j) \in \Omega$ is called an* irregular mesh point *if one or more of its four nearest neighbours lies in $R \setminus \bar{\Omega}$. A mesh point $(x_i, y_j) \in R \setminus \bar{\Omega}$ is called an* irregular mesh point *if one or more of its four nearest neighbours lies in Ω. All mesh points $(x_i, y_j) \in \Gamma$ are called* irregular mesh points. *Any mesh point which is not irregular is called a* regular mesh point.

As an example of this, in Figure 2-1 below, $L = 5$, $n = 31$ and Ω is the domain with boundary Γ described by the parametric representation

$$\gamma(t) = (\cos t(1 + \cos^2 4t), \sin t(1 + \cos^2 4t)), \quad t \in [0, 2\pi]. \tag{2.3.14}$$

Regular mesh points are given by dots, irregular mesh points in Ω are denoted by x and irregular mesh points in $R \setminus \bar{\Omega}$ are denoted by o. There are no mesh points on Γ. Note that the question of how to compute which mesh points are irregular is not an easy one to answer, especially for complicated domains. This problem will be addressed in §2.5.

Now there are an infinite number of particular solutions of (2.1.1), and each can be written as

$$u_p = U + V, \tag{2.3.15}$$

where U is the domain integral given by (2.1.6) and V satisfies

$$-\Delta V + \alpha^2 V = 0, \quad \text{in } \Omega.$$

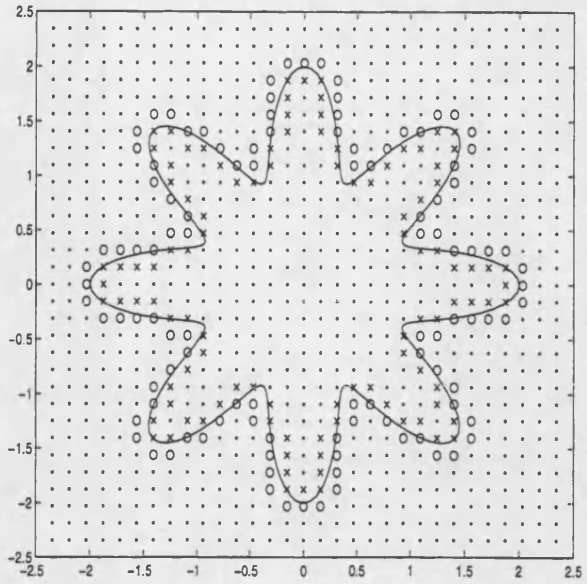

Figure 2-1: Irregular points inside (x) and outside (o) the domain with boundary curve (2.3.14).

In particular, if V is taken to be the solution of

$$-\Delta V + \alpha^2 V = 0 \quad \text{in } R \tag{2.3.16}$$

$$V = -U \quad \text{on } \partial R, \tag{2.3.17}$$

where ∂R is the boundary of R, then u_p given by (2.3.15) will be the particular solution of (2.1.1) corresponding to the boundary data $u_p = 0$ on ∂R, and with g extended by zero on $\mathbb{R}^2 \setminus \bar{\Omega}$.

To evaluate an approximation to u_p (given by (2.3.15)–(2.3.17)), the basic idea is then to compute an approximation $\tilde{f}_{i,j}$ to

$$f_{i,j} := -\Delta_h (u_p)_{i,j} + \alpha^2 (u_p)_{i,j}, \quad i,j = 1, \ldots, n, \tag{2.3.18}$$

where $(u_p)_{i,j} := u_p(x_i, y_j)$, and Δ_h is the five-point approximation to the Laplacian, given by

$$\Delta_h u_{i,j} := \frac{1}{h^2}(u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j} - 4u_{i,j}). \tag{2.3.19}$$

The main point is that although in principle the computation of u_p involves computing a domain integral, we can compute suitably accurate approximations $\tilde{f}_{i,j}$ to $f_{i,j}$ without doing *any* integration, as we shall demonstrate below.

Once we have computed $\tilde{f}_{i,j}$, we can then apply a fast Helmholtz solver on R_h to solve the linear system

$$-\Delta_h(\tilde{u}_p)_{i,j} + \alpha^2(\tilde{u}_p)_{i,j} = \tilde{f}_{i,j}, \quad \text{in } R_h, \tag{2.3.20}$$

supplemented by the boundary condition

$$(\tilde{u}_p)_{i,j} = 0, \quad \text{on } \partial R_h. \tag{2.3.21}$$

Combining (2.3.20), (2.3.18), (2.3.21), and the definition of u_p (2.3.15)–(2.3.17), we then have that

$$\begin{aligned}(-\Delta_h + \alpha^2)((u_p)_{i,j} - (\tilde{u}_p)_{i,j}) &= f_{i,j} - \tilde{f}_{i,j} \quad \text{in } R_h \\ ((u_p)_{i,j} - (\tilde{u}_p)_{i,j}) &= 0 \quad \text{on } \partial R_h.\end{aligned}$$

Using well known results on the stability of finite difference schemes (see for example [34, Theorem 4.4.11]) we get the error bound

$$\max_{i,j=1,\ldots,n} |(u_p)_{i,j} - (\tilde{u}_p)_{i,j}| \leq C \max_{i,j=1,\ldots,n} |f_{i,j} - \tilde{f}_{i,j}|, \tag{2.3.22}$$

where C is a constant independent of n.

Remark: If we specifically wanted to approximate the particular solution $U(\boldsymbol{x})$ given by (2.1.6) using this method, then we would need to supply the boundary values of U on ∂R. This could be done by explicitly computing the domain integral $U(\boldsymbol{x})$ for all $\boldsymbol{x} = (x_i, y_j) \in \partial R_h$. Although these integrals would be easier to evaluate than for $\boldsymbol{x} \in \Omega$, as the kernel would no longer be singular (since ∂R is outside Ω), they would still require the discretisation of Ω, and this could be expensive. However, in practice this is not necessary, as we are only seeking *any* particular solution of (2.1.1).

The question now is how to approximate $f_{i,j}$, and we devote the rest of this section to demonstrating how we can do this to second order accuracy. First we need the following elementary result (see for example [34, p.60]).

Lemma 2.3.3 *Let $v \in C^4(\bar{\Omega})$. Then for all regular points $(x_i, y_j) \in \bar{\Omega}$,*

$$\Delta_h v_{i,j} = \Delta v_{i,j} + h^2(R_x + R_y),$$

where $v_{i,j} = v(x_i, y_j)$,

$$|R_x|, |R_y| \leq \frac{1}{12}\|v\|_{4,\bar{\Omega}},$$

and $\|\cdot\|_{4,\bar{\Omega}}$ is the usual $C^4(\bar{\Omega})$ norm (see Appendix A).

Proof: Using a simple Taylor Series expansion, we have

$$\begin{aligned} v(x_{i\pm1}, y_j) &= v(x_i, y_j) \pm h v_x(x_i, y_j) + \frac{h^2}{2} v_{xx}(x_i, y_j) \\ &\quad \pm \frac{h^3}{6} v_{xxx}(x_i, y_j) + \frac{h^4}{24} v_{xxxx}(x_i \pm \theta h, y_j), \end{aligned}$$

where $\theta \in (0,1)$. We can derive a similar expression for $v(x_i, y_{j\pm1})$, and then adding them together and recalling the definition of Δ_h (2.3.19), the result follows. □

Now note that with $f_{i,j}$ defined by (2.3.18), we can write

$$f_{i,j} = (-\Delta(u_p)_{i,j} + \alpha^2(u_p)_{i,j}) + (\Delta(u_p)_{i,j} - \Delta_h(u_p)_{i,j}).$$

Considering the second term on the right hand side, we wish to see how well $\Delta_h(u_p)_{i,j}$ approximates $\Delta(u_p)_{i,j}$, and in order to do this we first need to consider the regularity properties of u_p. First we consider the domain integral U. In Appendix C we prove (in Theorem C.0.2) that if $\Gamma \in C^\infty$ and $g \in C^{l,\gamma}(\bar{\Omega})$, $\gamma \in (0,1)$ (this is the usual Hölder space - see Appendix A for a full definition), then $U \in C^{l+1,\gamma}(\bar{\Omega})$ and $U \in C^{l+1,\gamma}(\mathbb{R}^2 \setminus \Omega)$. This result is not obvious, as we explain in Appendix C, and indeed we have proved it there since we have been unable to find it anywhere in the literature, although we have found results closely approximating it (see for example [33], [43]).

Now consider the function V. This satisfies the homogeneous Helmholtz equation in the domain R, with boundary data given by $U|_{\partial R}$, and so V will be at least as smooth as U away from the corners of R, and it will not have any discontinuities across Γ.

Under the assumption that $g \in \mathcal{C}^{3,\gamma}(\bar{\Omega})$, for some $\gamma \in (0,1)$, we then have that

$$u_p \in \mathcal{C}^{4,\gamma}(\bar{\Omega}) \text{ and } u_p \in \mathcal{C}^{4,\gamma}(R \setminus \Omega). \tag{2.3.23}$$

So, at any regular point of the mesh, where we only need to take values of u_p from the same side of the boundary to form $\Delta_h(u_p)_{i,j}$, we can apply Lemma 2.3.3 directly. Using Definition 2.1.1 it is clear that

$$-\Delta u_p(\boldsymbol{x}) + \alpha^2 u_p(\boldsymbol{x}) = \begin{cases} g(\boldsymbol{x}), & \boldsymbol{x} \in \Omega \\ 0, & \boldsymbol{x} \in R \setminus \Omega, \end{cases} \tag{2.3.24}$$

where g is the function on the right hand side of (2.1.1). Applying Lemma 2.3.3 we then get

$$\begin{aligned} f_{i,j} &= (-\Delta + \alpha^2)(u_p)_{i,j} + (\Delta - \Delta_h)(u_p)_{i,j} \\ &= \begin{cases} g_{i,j} + \mathcal{O}(h^2), & \text{at regular points in } \Omega \\ 0 + \mathcal{O}(h^2), & \text{at regular points in } R \setminus \Omega, \end{cases} \end{aligned} \tag{2.3.25}$$

where $g_{i,j} = g(x_i, y_j)$.

At irregular points however we cannot use this simple formula. It is clear from (2.3.24) that the second order derivatives of u_p may be discontinuous across Γ. Hence, at irregular points, where the five-point Laplacian Δ_h involves values of u_p from both inside and outside Ω, $\Delta_h(u_p)_{i,j}$ will not be a good approximation to $\Delta(u_p)_{i,j}$. Thus we need a new method for determining an $\mathcal{O}(h^2)$ accurate approximation to $\Delta_h(u_p)_{i,j}$ at irregular points. This method breaks down into two parts.

In §2.3.1, we show how we can use Taylor series expansions to compute an $\mathcal{O}(h^2)$ accurate approximation to $\Delta_h(u_p)_{i,j}$, which involves the values of the discontinuities in u_p and its derivatives in the coordinate directions across Γ. This method was originally developed by Mayo in [58] and [61]. In §2.3.2 we show how we can use a method in [60] to actually compute these discontinuities. As well as describing these methods, we also fill in some details of the analysis which did not appear in the original papers.

Once we have an $\mathcal{O}(h^2)$ accurate approximation to $f_{i,j}$ at the irregular mesh

points, we can combine it with the formula at the regular mesh points (2.3.25) to get a linear system of the form (2.3.20). The question of how to efficiently invert the discrete Helmholtz operator to solve (2.3.20), (2.3.21) will be addressed in §2.4.

2.3.1 Approximating $f_{i,j}$ at irregular points

Throughout this section we assume that g in (2.1.6) satisfies $g \in \mathcal{C}^{3,\gamma}(\bar{\Omega})$, for some $\gamma \in (0,1)$, which ensures that (2.3.23) holds and thus that $u_p(\boldsymbol{x})$ has four continuous derivatives at every point $\boldsymbol{x} \in R \setminus \Gamma$, and, although these derivatives can jump across Γ, they are well behaved as $\boldsymbol{x}$ approaches Γ from either the inside or the outside.

We then seek an approximation to $\Delta_h(u_p)_{i,j}$ at the irregular mesh points. Consider the example in Figure 2-2 below, where $P := (x_i, y_j)$ is an irregular mesh point in Ω, $P_w := (x_{i-1}, y_j)$ and $P_s := (x_i, y_{j-1})$ lie in Ω, $P_e := (x_{i+1}, y_j)$ and $P_n := (x_i, y_{j+1})$ lie in $R \setminus \Omega$, P_e^* is the point where Γ crosses the line from P to P_e, P_n^* is the point where Γ crosses the line from P to P_n, h_e is the distance from P to P_e^*, h'_e is the distance from P_e^* to P_e, h_n is the distance from P to P_n^*, and h'_n is the distance from P_n^* to P_n. Then

$$h_e + h'_e = h_n + h'_n = h.$$

If Γ crossed the lines from P to P_w or from P to P_s, then the points where the lines crossed would be P_w^* and P_s^* respectively, and h_w, h'_w, h_s and h'_s would be defined analogously to h_e, h'_e, h_n and h'_n.

We assume throughout that Γ does not cross any of the lines from P to P_n, P to P_s, P to P_e or P to P_w more than once. If this were not the case, and Γ crossed the line from (for example) P to P_e exactly twice, then P_e would no longer be outside Ω, and so (provided P_s, P_n and P_w also were in Ω) P would then be a regular point. In theory it is possible that Γ could cross one of the lines three or even more times, but this is a slightly pathological scenario and it is reasonable to assume that this cannot occur for h sufficiently small. These cases can be identified provided γ is locally invertible, and a similar analysis to that described below can be performed.

The idea in the case presented in Figure 2-2 is to work out the values of

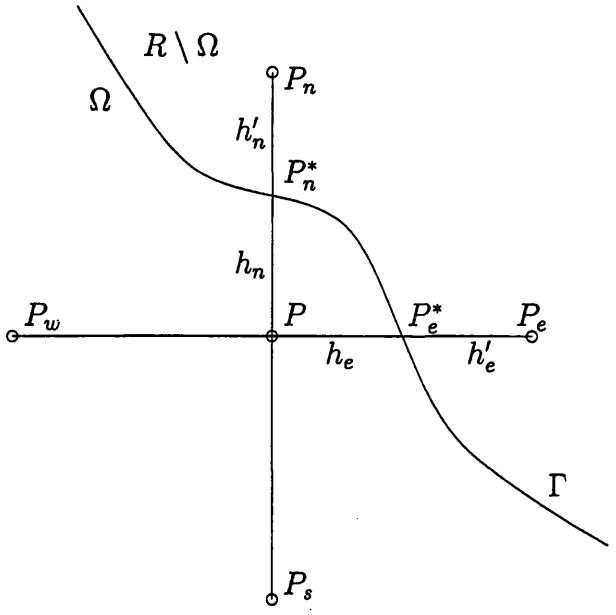

Figure 2-2: Approximating $\Delta_h(u_p)_{i,j}$ at the irregular point $P = (x_i, y_j)$.

$\{u_p(P_e) - u_p(P)\}$ and $\{u_p(P_n) - u_p(P)\}$ in terms of the jumps in the values of u_p and its derivatives across Γ. Having done this, it turns out to be simple to construct an approximation to $\Delta_h(u_p)_{i,j}$ in terms of these jumps. First, we show how we can do this for $\{u_p(P_e) - u_p(P)\}$.

Lemma 2.3.4 *For any function v satisfying $v \in \mathcal{C}^4(\bar{\Omega})$ and $v \in \mathcal{C}^4(R \setminus \Omega)$, and with the mesh points $P = (x_i, y_j) \in \Omega$ and $P_e = (x_{i+1}, y_j) \notin \Omega$, we have the formula*

$$\begin{aligned} v(P_e) - v(P) &= hv_x(P) + \frac{h^2}{2}v_{xx}(P) + \frac{h^3}{6}v_{xxx}(P) \\ &\quad - \left\{ [v(P^*_e)] + h'_e[v_x(P^*_e)] + \frac{h'^2_e}{2}[v_{xx}(P^*_e)] + \frac{h'^3_e}{6}[v_{xxx}(P^*_e)] \right\} \\ &\quad + \mathcal{O}(h^4), \end{aligned}$$

where for any function $f \in \mathcal{C}(R \setminus \Gamma)$ the function $[f] : \Gamma \mapsto \mathbb{R}$ is defined by

$$[f(\boldsymbol{x})] := f^-(\boldsymbol{x}) - f^+(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma,$$

and the functions $f^- : \Gamma \mapsto \mathbb{R}$ *and* $f^+ : \Gamma \mapsto \mathbb{R}$ *are defined by*

$$\begin{aligned} f^-(\boldsymbol{x}) &:= \lim_{\Omega \ni \boldsymbol{y} \to \boldsymbol{x}} f(\boldsymbol{y}), \quad \boldsymbol{x} \in \Gamma, \\ f^+(\boldsymbol{x}) &:= \lim_{R \setminus \Omega \ni \boldsymbol{y} \to \boldsymbol{x}} f(\boldsymbol{y}), \quad \boldsymbol{x} \in \Gamma, \end{aligned}$$

provided these limits exist.

Proof: With P, P_e, P_e^*, h_e and h_e' as in Figure 2-2, we can use the usual Taylor series expansion of $v|_{\bar{\Omega}}$ about P to show that there exists $\theta_1 \in (0,1)$ such that

$$v^-(P_e^*) = v(P) + h_e v_x(P) + \frac{h_e^2}{2} v_{xx}(P) + \frac{h_e^3}{6} v_{xxx}(P) + \frac{h_e^4}{24} v_{xxxx}(P + \theta_1 h_e),$$

and hence

$$v(P) = v^-(P_e^*) - h_e v_x(P) - \frac{h_e^2}{2} v_{xx}(P) - \frac{h_e^3}{6} v_{xxx}(P) - \frac{h_e^4}{24} v_{xxxx}(P + \theta_1 h_e). \tag{2.3.26}$$

Similarly, expanding $v|_{R\setminus\Omega}$ about P_e shows that there exists $\theta_2 \in (0,1)$ such that

$$v(P_e) = v^+(P_e^*) + h_e' v_x(P_e) - \frac{h_e'^2}{2} v_{xx}(P_e) + \frac{h_e'^3}{6} v_{xxx}(P_e) - \frac{h_e'^4}{24} v_{xxxx}(P_e - \theta_2 h_e'). \tag{2.3.27}$$

Hence

$$\begin{aligned} v(P_e) - v(P) &= (v^+(P_e^*) - v^-(P_e^*)) + h_e' v_x(P_e) + h_e v_x(P) \\ &\quad - \frac{h_e'^2}{2} v_{xx}(P_e) + \frac{h_e^2}{2} v_{xx}(P) + \frac{h_e'^3}{6} v_{xxx}(P_e) + \frac{h_e^3}{6} v_{xxx}(P) \\ &\quad - \frac{h_e'^4}{24} v_{xxxx}(P_e - \theta_2 h_e') + \frac{h_e^4}{24} v_{xxxx}(P + \theta_1 h_e). \end{aligned} \tag{2.3.28}$$

We now wish to express the right hand side of (2.3.28) in terms of $[v(P_e^*)]$, $[v_x(P_e^*)]$, $[v_{xx}(P_e^*)]$ and $[v_{xxx}(P_e^*)]$. It is clear that the $\mathcal{O}(1)$ term on the right hand side of (2.3.28) is $-[v(P_e^*)]$. So, consider the $\mathcal{O}(h)$ term. First note that

$$\begin{aligned} h_e v_x(P) &= (h - h_e') v_x(P) \\ &= h v_x(P) - h_e' v_x^-(P_e^*) + h_e'(v_x^-(P_e^*) - v_x(P)). \end{aligned}$$

Using a Taylor series expansion of $v_x|_{R\setminus\Omega}$ about P_e and rearranging the terms we also get, for some $\theta_3 \in (0,1)$,

$$v_x(P_e) = v_x^+(P_e^*) + h'_e v_{xx}(P_e) - \frac{h'^2_e}{2} v_{xxx}(P_e) + \frac{h'^3_e}{6} v_{xxxx}(P_e - \theta_3 h'_e).$$

Hence the $\mathcal{O}(h)$ term in (2.3.28) will be

$$\begin{aligned} h'_e v_x(P_e) + h_e v_x(P) &= h v_x(P) - h'_e[v_x(P_e^*)] + h'_e(v_x^-(P_e^*) - v_x(P)) \\ &\quad + h'^2_e v_{xx}(P_e) - \frac{h'^3_e}{2} v_{xxx}(P_e) + \frac{h'^4_e}{6} v_{xxxx}(P_e - \theta_3 h'_e). \end{aligned} \tag{2.3.29}$$

Expanding $v_x|_\Omega$ about P we see that

$$v_x^-(P_e^*) - v_x(P) = h_e v_{xx}(P) + \frac{h_e^2}{2} v_{xxx}(P) + \frac{h_e^3}{6} v_{xxxx}(P + \theta_4 h_e),$$

for some $\theta_4 \in (0,1)$, and then substituting this into the right hand side of (2.3.29) we get

$$\begin{aligned} h'_e v_x(P_e) + h_e v_x(P) &= h v_x(P) - h'_e[v_x(P_e^*)] + h_e h'_e v_{xx}(P) + h'^2_e v_{xx}(P_e) \\ &\quad + \frac{h_e^2 h'_e}{2} v_{xxx}(P) - \frac{h'^3_e}{2} v_{xxx}(P_e) + \mathcal{O}(h^4). \end{aligned} \tag{2.3.30}$$

Looking at the $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$ terms in a similar way, we can derive the required result, after much tedious algebra. □

Remark: Note that if v is smooth across Γ, then this formula reduces to the usual Taylor series expansion,

$$v(P_e) - v(P) = h v_x(P) + \frac{h^2}{2} v_{xx}(P) + \frac{h^3}{6} v_{xxx}(P) + \mathcal{O}(h^4),$$

since in this case all of the jumps will be zero.

So, Lemma 2.3.4 gives us a formula for $\{u_p(P_e) - u_p(P)\}$, and we can derive similar formulae for $\{u_p(P_w) - u_p(P)\}$, $\{u_p(P_n) - u_p(P)\}$, and $\{u_p(P_s) - u_p(P)\}$, as are given in the following lemma.

Lemma 2.3.5 *For any function v satisfying $v \in C^4(\bar{\Omega})$ and $v \in C^4(R \setminus \Omega)$, we have the following formulae.*

i). If $P \in \Omega$ and $P_w \notin \Omega$ then

$$
\begin{aligned}
& v(P_w) - v(P) = \\
& \quad -hv_x(P) + \frac{h^2}{2}v_{xx}(P) - \frac{h^3}{6}v_{xxx}(P) \\
& \quad - \left\{ [v(P_w^*)] - h'_w[v_x(P_w^*)] + \frac{h'^{\,2}_w}{2}[v_{xx}(P_w^*)] - \frac{h'^{\,3}_w}{6}[v_{xxx}(P_w^*)] \right\} \\
& \quad + \mathcal{O}(h^4).
\end{aligned}
$$

ii). If $P \in \Omega$ and $P_n \notin \Omega$ then

$$
\begin{aligned}
& v(P_n) - v(P) = \\
& \quad hv_y(P) + \frac{h^2}{2}v_{yy}(P) + \frac{h^3}{6}v_{yyy}(P) \\
& \quad - \left\{ [v(P_n^*)] + h'_n[v_y(P_n^*)] + \frac{h'^{\,2}_n}{2}[v_{yy}(P_n^*)] + \frac{h'^{\,3}_n}{6}[v_{yyy}(P_n^*)] \right\} \\
& \quad + \mathcal{O}(h^4).
\end{aligned}
$$

iii). If $P \in \Omega$ and $P_s \notin \Omega$ then

$$
\begin{aligned}
& v(P_s) - v(P) = \\
& \quad -hv_y(P) + \frac{h^2}{2}v_{yy}(P) - \frac{h^3}{6}v_{yyy}(P) \\
& \quad - \left\{ [v(P_s^*)] - h'_s[v_y(P_s^*)] + \frac{h'^{\,2}_s}{2}[v_{yy}(P_s^*)] - \frac{h'^{\,3}_s}{6}[v_{yyy}(P_s^*)] \right\} \\
& \quad + \mathcal{O}(h^4).
\end{aligned}
$$

Proof: The proofs are similar to the proof of Lemma 2.3.4. □

Now we are in a position to derive a formula for the five-point Laplacian $\Delta_h(u_p)_{i,j}$ at irregular points in Ω. In order to make the formula general for any irregular point P in Ω, we define $[v(P_k^*)]$ to be zero if P_k^* does not exist, ie. if P_k also lies in Ω (where $k = n, s, e, w$).

Lemma 2.3.6 *For any function v satisfying $v \in \mathcal{C}^4(\bar{\Omega})$ and $v \in \mathcal{C}^4(R \setminus \Omega)$, and*

for all irregular points $P = (x_i, y_j) \in \Omega$, *define*

$$\begin{aligned} c(P) \;:=\; & \frac{1}{h^2}([v(P_n^*)] + [v(P_s^*)] + [v(P_e^*)] + [v(P_w^*)]) \\ & + \frac{1}{h^2}(h_n'[v_y(P_n^*)] - h_s'[v_y(P_s^*)] + h_e'[v_x(P_e^*)] - h_w'[v_x(P_w^*)]) \\ & + \frac{1}{2h^2}(h_n'^{\,2}[v_{yy}(P_n^*)] + h_s'^{\,2}[v_{yy}(P_s^*)] + h_e'^{\,2}[v_{xx}(P_e^*)] + h_w'^{\,2}[v_{xx}(P_w^*)]) \\ & + \frac{1}{6h^2}(h_n'^{\,3}[v_{yyy}(P_n^*)] - h_s'^{\,3}[v_{yyy}(P_s^*)] \\ & \qquad + h_e'^{\,3}[v_{xxx}(P_e^*)] - h_w'^{\,3}[v_{xxx}(P_w^*)]). \end{aligned} \tag{2.3.31}$$

Then at all irregular points in Ω,

$$\Delta_h v_{i,j} = \Delta v_{i,j} - c_{i,j} + \mathcal{O}(h^2), \tag{2.3.32}$$

where $v_{i,j} := v(x_i, y_j)$ *and* $c_{i,j} := c(x_i, y_j) = c(P)$.

Proof: From the formula (2.3.19) for the five-point Laplacian, we have

$$\begin{aligned} & \Delta_h v_{i,j} = \\ & \quad \frac{1}{h^2}(\{v(P_n) - v(P)\} + \{v(P_s) - v(P)\} + \{v(P_e) - v(P)\} + \{v(P_w) - v(P)\}), \end{aligned}$$

and using the formulae of Lemmas 2.3.4 and 2.3.5 the result follows easily. □

Finally, we derive an analogous formula for irregular points in $R \setminus \Omega$.

Lemma 2.3.7 *For any function* v *satisfying* $v \in \mathcal{C}^4(\bar{\Omega})$ *and* $v \in \mathcal{C}^4(R \setminus \Omega)$, *and for all irregular points* $P = (x_i, y_j) \in R \setminus \Omega$, *define* $c(P)$ *as in Lemma 2.3.6. Then at all irregular points in* $R \setminus \Omega$,

$$\Delta_h v_{i,j} = \Delta v_{i,j} + c_{i,j} + \mathcal{O}(h^2). \tag{2.3.33}$$

Proof: With $P \in R \setminus \Omega$, we can derive similar formulae to those in Lemmas 2.3.4 and 2.3.5 for $\{v(P_k) - v(P)\}$, $k = n, s, e, w$, but where we had $(v^-(P_j^*) - v^+(P_j^*))$ before, we will now have $(v^+(P_j^*) - v^-(P_j^*))$. Hence all of the jump discontinuities will have the opposite sign to before, and following the proof through analogously to the case when $P \in \Omega$, the result follows. □

Using the results from Lemmas 2.3.6 and 2.3.7, with u_p taking the place of v, and recalling (2.3.24) and (2.3.18), we get

$$\begin{aligned} f_{i,j} &= (-\Delta + \alpha^2)(u_p)_{i,j} + (\Delta - \Delta_h)(u_p)_{i,j} \\ &= \begin{cases} g_{i,j} + c_{i,j} + \mathcal{O}(h^2), & \text{at irregular points in } \Omega \\ -c_{i,j} + \mathcal{O}(h^2), & \text{at irregular points in } R \setminus \Omega. \end{cases} \end{aligned} \tag{2.3.34}$$

We thus have an $\mathcal{O}(h^2)$ accurate approximation to the discrete Helmholtz operator at all irregular mesh points in R, in terms of the unknown jump discontinuities in u_p and its derivatives in the coordinate directions across Γ. In the next section we show how to evaluate these jump discontinuities.

2.3.2 Evaluating the discontinuites across Γ.

Taking $u_p = U + V$ as in (2.3.15), with U given by (2.1.6) and V given by (2.3.16), (2.3.17), it is apparent that because V has no discontinuities across Γ, as explained in §2.3, we need only compute the discontinuities in the domain integral U and its derivatives across Γ. First, we need the following result.

Lemma 2.3.8 *If $\Gamma \in C^2$, and if $g \in L_\infty(\Omega)$, then the volume integral U and its first order derivatives in every direction are all continuous across Γ.*

Proof: Let $g \in L_\infty(\Omega)$, let D be any domain containing Ω, so that the boundary of Ω lies inside the boundary of D, and define $\tilde{g}$ on D by extending g by zero outside Ω. Then $\tilde{g} \in L_\infty(D)$. Define $\mathcal{P}_D$ thus:

$$\mathcal{P}_D \tilde{g}(\boldsymbol{x}) := \int_D \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) \tilde{g}(\boldsymbol{y}) d\boldsymbol{y} = \int_\Omega \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{y}) d\boldsymbol{y} = U(\boldsymbol{x}).$$

The first part of the proof of Theorem C.0.10 indicates that $\mathcal{P}_D \tilde{g} \in C^{1,\gamma}(D)$, $\gamma \in (0,1)$, and so $U \in C^{1,\gamma}(D)$, as required. □

Now for any point $\boldsymbol{x} \in \Gamma$ we write

$$\boldsymbol{x} = \boldsymbol{\gamma}(t) = (\gamma_1(t), \gamma_2(t)), \quad \text{for some } t \in [0, 2\pi],$$

where $\gamma : [0, 2\pi] \mapsto \Gamma$ is a 2π-periodic parametrisation of Γ satisfying (1.1.5) and (1.1.7). Using Lemma 2.3.8 we then have, for $\boldsymbol{x} \in \Gamma$,

$$0 = \left[\frac{\partial U}{dn}(\boldsymbol{x})\right] = \gamma_2'(t)[U_x(\boldsymbol{x})] - \gamma_1'(t)[U_y(\boldsymbol{x})], \tag{2.3.35}$$

and

$$0 = \left[\frac{\partial U}{dt}(\boldsymbol{x})\right] = \gamma_1'(t)[U_x(\boldsymbol{x})] + \gamma_2'(t)[U_y(\boldsymbol{x})]. \tag{2.3.36}$$

Now clearly from Lemma 2.3.8 we also have

$$[U_x(\boldsymbol{x})] = [U_y(\boldsymbol{x})] = 0, \quad \text{for all } \boldsymbol{x} \in \Gamma, \tag{2.3.37}$$

but note that we can also derive (2.3.37) by combining (2.3.35) and (2.3.36) to get

$$\begin{pmatrix} \gamma_2'(t) & -\gamma_1'(t) \\ \gamma_1'(t) & \gamma_2'(t) \end{pmatrix} \begin{pmatrix} [U_x(\boldsymbol{x})] \\ [U_y(\boldsymbol{x})] \end{pmatrix} = 0, \tag{2.3.38}$$

and then (2.3.37) follows since the determinant of the coefficient matrix is $\gamma_1'(t)^2 + \gamma_2'(t)^2$, which does not vanish by (1.1.7).

Now from (2.3.24) it is clear that U can have discontinuities in its second-order derivatives. To find these, we first differentiate (2.3.35) in the tangential direction, and then using (2.3.37) we get (remembering that $\boldsymbol{x} = \gamma(t)$)

$$\begin{aligned} 0 &= \frac{\partial}{\partial t}\{\gamma_2'(t)[U_x(\boldsymbol{x})] - \gamma_1'(t)[U_y(\boldsymbol{x})]\} \\ &= \gamma_2''(t)\,[U_x(\boldsymbol{x})] + \gamma_2'(t)\{\gamma_1'(t)\,[U_{xx}(\boldsymbol{x})] + \gamma_2'(t)\,[U_{xy}(\boldsymbol{x})]\} \\ &\quad - \gamma_1''(t)\,[U_y(\boldsymbol{x})] - \gamma_1'(t)\{\gamma_1'(t)\,[U_{xy}(\boldsymbol{x})] + \gamma_2'(t)\,[U_{yy}(\boldsymbol{x})]\} \\ &= \gamma_1'(t)\gamma_2'(t)\,[U_{xx}(\boldsymbol{x})] + (\gamma_2'(t)^2 - \gamma_1'(t)^2)\,[U_{xy}(\boldsymbol{x})] - \gamma_1'(t)\gamma_2'(t)\,[U_{yy}(\boldsymbol{x})]\,. \end{aligned} \tag{2.3.39}$$

Similarly, differentiating (2.3.36) in the tangential direction, and using (2.3.37) we get

$$\gamma_1'(t)^2\,[U_{xx}(\boldsymbol{x})] + 2\gamma_1'(t)\gamma_2'(t)\,[U_{xy}(\boldsymbol{x})] + \gamma_2'(t)^2\,[U_{yy}(\boldsymbol{x})] = 0. \tag{2.3.40}$$

Now from Definition 2.1.1 we know that U satisfies

$$-U_{xx}(\boldsymbol{x}) - U_{yy}(\boldsymbol{x}) + \alpha^2 U(\boldsymbol{x}) = \begin{cases} g(\boldsymbol{x}) & \boldsymbol{x} \in \Omega \\ 0 & \boldsymbol{x} \in R \setminus \Omega, \end{cases} \tag{2.3.41}$$

and so

$$[U_{xx}(\boldsymbol{x})] + [U_{yy}(\boldsymbol{x})] = -g(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma. \tag{2.3.42}$$

Combining (2.3.39), (2.3.40) and (2.3.42) we get

$$\begin{pmatrix} {\gamma_1}'(t)^2 & 2{\gamma_1}'(t){\gamma_2}'(t) & {\gamma_2}'(t)^2 \\ {\gamma_1}'(t){\gamma_2}'(t) & {\gamma_2}'(t)^2 - {\gamma_1}'(t)^2 & -{\gamma_1}'(t){\gamma_2}'(t) \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} [U_{xx}(\boldsymbol{x})] \\ [U_{xy}(\boldsymbol{x})] \\ [U_{yy}(\boldsymbol{x})] \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -g(\boldsymbol{x}) \end{pmatrix}. \tag{2.3.43}$$

The matrix on the left hand side of (2.3.43) can be shown to have determinant equal to $-({\gamma_1}'(t)^2 + {\gamma_2}'(t)^2)^2$, which is non-zero by (1.1.7). Hence the system (2.3.43) has a unique solution, and can be solved for the jumps in the second order derivatives of U.

To find the jumps in the third-order derivatives of U in the coordinate directions we follow a similar procedure. Differentiating (2.3.39) and (2.3.40) in the tangential direction, and differentiating (2.3.41) in the coordinate directions and considering the jumps across Γ, we get a linear system of the form

$$A\boldsymbol{u} = \boldsymbol{b}, \tag{2.3.44}$$

where

$$A = \begin{pmatrix} {\gamma_1}'(t)^3 & 3{\gamma_1}'(t)^2{\gamma_2}'(t) & 3{\gamma_1}'(t){\gamma_2}'(t)^2 & {\gamma_2}'(t)^3 \\ {\gamma_1}'(t)^2{\gamma_2}'(t) & a_{2,2} & a_{2,3} & -{\gamma_1}'(t){\gamma_2}'(t)^2 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix},$$

with $a_{2,2} = 2{\gamma_1}'(t){\gamma_2}'(t)^2 - {\gamma_1}'(t)^3$ and $a_{2,3} = {\gamma_2}'(t)^3 - 2{\gamma_1}'(t)^2{\gamma_2}'(t)$,

$$\boldsymbol{u} = \begin{pmatrix} [U_{xxx}(\boldsymbol{x})] & [U_{xxy}(\boldsymbol{x})] & [U_{xyy}(\boldsymbol{x})] & [U_{yyy}(\boldsymbol{x})] \end{pmatrix}^T,$$

and

$$\boldsymbol{b} = \begin{pmatrix} -2\gamma_1'(t)\gamma_1''(t)\,[U_{xx}(\boldsymbol{x})] - 2(\gamma_1'(t)\gamma_2''(t) + \gamma_1''(t)\gamma_2'(t))\,[U_{xy}(\boldsymbol{x})] \\ \quad - 2\gamma_2'(t)\gamma_2''(t)\,[U_{yy}(\boldsymbol{x})] \\ (\gamma_1'(t)\gamma_2''(t) + \gamma_1''(t)\gamma_2'(t))([U_{yy}(\boldsymbol{x})] - [U_{xx}(\boldsymbol{x})]) \\ \quad - 2(\gamma_2'(t)\gamma_2''(t) - \gamma_1'(t)\gamma_1''(t))\,[U_{xy}(\boldsymbol{x})] \\ -\partial g/\partial x \\ -\partial g/\partial y \end{pmatrix}.$$

It can be shown that $|A| = (\gamma_1'(t)^2 + \gamma_2'(t)^2)^3$, which again is non-zero by (1.1.7). Hence this system also has a unique solution.

By repeatedly differentiating (2.3.39) and (2.3.40) in the tangential direction, and (2.3.41) in the coordinate directions, we could derive larger and larger systems for the higher order jump discontinuities. It is tempting to conjecture that the matrix on the left hand side of the system for the unknown discontinuities of order k will have determinant equal to $(-1)^{k+1}(\gamma_1'(t)^2 + \gamma_2'(t)^2)^k$, as this is the case for $k = 1, 2, 3$ in (2.3.38), (2.3.43) and (2.3.44). However, we have been unable to prove this for a general value of k.

We can now use (2.3.43) and (2.3.44) to determine the unknown jump discontinuities in the second and third-order derivatives. We can then use these in the formulae of Lemmas 2.3.6 and 2.3.7 to evaluate the correction term near the boundary.

In practice, in order to form the right hand side of (2.3.44) one needs to know the derivatives of the inhomogeneous term g. If g is known only as data, then these derivatives can be approximated using some numerical differentiation scheme, but an order of accuracy will be lost. However, this is not a problem for us as we only need to compute the third order jump discontinuities to $\mathcal{O}(h)$ accuracy in order for the formulae of Lemmas 2.3.6 and 2.3.7 to hold. Indeed, it is claimed in [62] that one does not need to compute the third order jump discontinuities at all, and that merely to approximate the five-point Laplacian to $\mathcal{O}(h)$ accuracy at the irregular points is sufficient. However, we have been unable to find a proof of this assertion.

One last thing to note is that the matrices in (2.3.43) and (2.3.44), which need inverting in order to find the jump discontinuites, are dependent only on the shape of the domain. Hence if one needs to repeat this procedure for several

volume integrals with different densities, as will be the case in the context of our complete method for solving parabolic PDEs, the matrices and their inverses only need to be computed once.

2.4 Solving the extended problem

We are now in a position to solve the linear system (2.3.20)–(2.3.21), which we repeat here for clarity

$$-\Delta_h(\tilde{u}_p)_{i,j} + \alpha^2(\tilde{u}_p)_{i,j} = \tilde{f}_{i,j}, \quad \text{in } R_h, \tag{2.4.45}$$

$$(\tilde{u}_p)_{i,j} = 0, \quad \text{on } \partial R_h. \tag{2.4.46}$$

Combining the approximations to $f_{i,j} = (-\Delta_h + \alpha^2)(u_p)_{i,j}$ at regular (2.3.25) and irregular (2.3.34) points, where $(u_p)_{i,j} = u_p(x_i, y_j)$, and u_p is the particular solution given by (2.3.15) (where U is given by (2.1.6) and V is the solution of (2.3.16)–(2.3.17)), we see that if we choose

$$\tilde{f}_{i,j} = \begin{cases} g_{i,j} & \text{at regular points in } \Omega \\ 0 & \text{at regular points in } R \setminus \Omega \\ g_{i,j} + c_{i,j} & \text{at irregular points in } \Omega \\ -c_{i,j} & \text{at irregular points in } R \setminus \Omega, \end{cases} \tag{2.4.47}$$

then this satisfies

$$\tilde{f}_{i,j} = f_{i,j} + \mathcal{O}(h^2).$$

Using the error bound (2.3.22), we then have the following result.

Theorem 2.4.1 *For all $g \in \mathcal{C}^{3,\gamma}(\bar{\Omega})$, $\gamma \in (0,1)$, and provided Γ is parametrised by a 2π-periodic function $\boldsymbol{\gamma} : \mathbb{R} \mapsto \Gamma$ which satisfies (1.1.5)–(1.1.7), there exists a constant C independent of h such that*

$$\max_{i,j=1,\ldots,n} |(u_p)_{i,j} - (\tilde{u}_p)_{i,j}| \leq Ch^2, \tag{2.4.48}$$

where u_p is given by (2.3.15) and $\tilde{u}_p$ is the solution of (2.4.45)–(2.4.46).

The question now is how to solve (2.4.45)–(2.4.46) efficiently. In this section we show how Fast Fourier Transform (FFT) methods can be used to do this. The number of operations needed for the computation of the n^2 entries $(\tilde{u}_p)_{i,j}$ is $\mathcal{O}(n^2 \log n)$.

As we have Dirichlet boundary data (2.4.46), here we solve (2.4.45) using a Fast Sine Transform method. If we had Neumann boundary data, we would instead use a Fast Cosine Transform method, see [80] or [67] for details.

First, we multiply both sides of (2.4.45) by h^2, and then recalling the formula for the five-point Laplacian (2.3.19), we get the $n^2 \times n^2$ system

$$A\tilde{\boldsymbol{u}} = h^2 \tilde{\boldsymbol{f}}, \tag{2.4.49}$$

where

$$A = \begin{pmatrix} T & -I & 0 & \cdot & 0 \\ -I & T & -I & \cdot & 0 \\ 0 & -I & T & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & -I \\ 0 & 0 & 0 & -I & T \end{pmatrix},$$

$$T = \begin{pmatrix} 4+\alpha^2h^2 & -1 & 0 & \cdot & 0 \\ -1 & 4+\alpha^2h^2 & -1 & \cdot & 0 \\ 0 & -1 & 4+\alpha^2h^2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & -1 \\ 0 & 0 & 0 & -1 & 4+\alpha^2h^2 \end{pmatrix},$$

I is the $n \times n$ identity matrix,

$$\tilde{\boldsymbol{u}} = ((\tilde{u}_p)_{1,1}, \ldots, (\tilde{u}_p)_{1,n}, (\tilde{u}_p)_{2,1}, \ldots, (\tilde{u}_p)_{2,n}, \longleftrightarrow (\tilde{u}_p)_{n,1}, \ldots, (\tilde{u}_p)_{n,n})^T,$$

and

$$\tilde{\boldsymbol{f}} = \left(\tilde{f}_{1,1}, \ldots, \tilde{f}_{1,n}, \tilde{f}_{2,1}, \ldots, \tilde{f}_{2,n}, \longleftrightarrow \tilde{f}_{n,1}, \ldots, \tilde{f}_{n,n}\right)^T.$$

Now, it can be shown that T has eigenvectors

$$\boldsymbol{v}_j = (\sin \pi jh, \sin 2\pi jh, \sin 3\pi jh, \cdots, \sin n\pi jh)^T, \quad j = 1, \ldots, n,$$

with corresponding eigenvalues

$$\lambda_j = 4 + \alpha^2 h^2 - 2\cos\frac{\pi jh}{L}, \quad j = 1, \ldots, n,$$

where L is the side length of the square R. The eigenvectors $\boldsymbol{v}_j$ are the columns of an $n \times n$ "sine matrix" S, which diagonalises T, ie. $S^{-1}TS = \bigwedge$, a diagonal matrix with the eigenvalues λ_j on the diagonal.

So, we now know that we can diagonalise T using S, and we want to do something similar to the $n^2 \times n^2$ matrix A. First, we define

$$\tilde{\boldsymbol{u}}_k = \tilde{\boldsymbol{u}}(1 + (k-1)n : kn) = ((\tilde{u}_p)_{k,1}, \ldots, (\tilde{u}_p)_{k,n})^T, \quad k = 1, \ldots, n,$$

and

$$\tilde{\boldsymbol{f}}_k = \tilde{\boldsymbol{f}}(1 + (k-1)n : kn) = \left(\tilde{f}_{k,1}, \ldots, \tilde{f}_{k,n}\right)^T, \quad k = 1, \ldots, n.$$

Then we can write (2.4.49) as

$$\begin{aligned} T\tilde{\boldsymbol{u}}_1 - \tilde{\boldsymbol{u}}_2 &= h^2\tilde{\boldsymbol{f}}_1 \\ -\tilde{\boldsymbol{u}}_{k-1} + T\tilde{\boldsymbol{u}}_k - \tilde{\boldsymbol{u}}_{k+1} &= h^2\tilde{\boldsymbol{f}}_k \quad k = 2, \ldots, n-1 \\ -\tilde{\boldsymbol{u}}_{n-1} + T\tilde{\boldsymbol{u}}_n &= h^2\tilde{\boldsymbol{f}}_n. \end{aligned}$$

Multiplying each side by S^{-1}, and recalling that $S^{-1}TS = \bigwedge$, we get

$$\begin{aligned} \bigwedge S^{-1}\tilde{\boldsymbol{u}}_1 - S^{-1}\tilde{\boldsymbol{u}}_2 &= h^2 S^{-1}\tilde{\boldsymbol{f}}_1 \\ -S^{-1}\tilde{\boldsymbol{u}}_{k-1} + \bigwedge S^{-1}\tilde{\boldsymbol{u}}_k - S^{-1}\tilde{\boldsymbol{u}}_{k+1} &= h^2 S^{-1}\tilde{\boldsymbol{f}}_k \quad k = 2, \ldots, n-1 \\ -S^{-1}\tilde{\boldsymbol{u}}_{n-1} + \bigwedge S^{-1}\tilde{\boldsymbol{u}}_n &= h^2 S^{-1}\tilde{\boldsymbol{f}}_n. \end{aligned}$$

Now writing $S^{-1}\tilde{\boldsymbol{u}}_k = \boldsymbol{p}_k$ and $S^{-1}\tilde{\boldsymbol{f}}_k = \boldsymbol{q}_k$, we have the $n^2 \times n^2$ system

$$\begin{pmatrix} \Lambda & -I & 0 & \cdot & 0 \\ -I & \Lambda & -I & \cdot & 0 \\ 0 & -I & \Lambda & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & -I \\ 0 & 0 & 0 & -I & \Lambda \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \\ \cdot \\ \boldsymbol{p}_n \end{pmatrix} = h^2 \begin{pmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \boldsymbol{q}_3 \\ \cdot \\ \boldsymbol{q}_n \end{pmatrix}. \tag{2.4.50}$$

Taking out the j^{th} component of each $\boldsymbol{p}_k$ and $\boldsymbol{q}_k$ we get the n tridiagonal $n \times n$ systems

$$\begin{pmatrix} \lambda_j & -1 & 0 & \cdot & 0 \\ -1 & \lambda_j & -1 & \cdot & 0 \\ 0 & -1 & \lambda_j & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & -1 \\ 0 & 0 & 0 & -1 & \lambda_j \end{pmatrix} \begin{pmatrix} p_{1,j} \\ p_{2,j} \\ p_{3,j} \\ \cdot \\ p_{n,j} \end{pmatrix} = h^2 \begin{pmatrix} q_{1,j} \\ q_{2,j} \\ q_{3,j} \\ \cdot \\ q_{n,j} \end{pmatrix}, \quad j = 1, \ldots, n. \tag{2.4.51}$$

Now for each $j = 1, \ldots, n$, these systems are decoupled from each other. Hence instead of having to solve an $n^2 \times n^2$ system, we now just have to solve n tridiagonal $n \times n$ systems. These can each be solved easily using a fast tridiagonal solver, such as the Thomas method (Gaussian elimination, exploiting the fact that the matrix is tridiagonal). The solutions of these n tridiagonal systems fill out the vectors $\boldsymbol{p}_k$, from which we recover the original unknowns $\tilde{\boldsymbol{u}}_k = S\boldsymbol{p}_k$. So, in summary the four steps of the method are:-

i). Compute the n-vector $\boldsymbol{q}_k = S^{-1}\tilde{\boldsymbol{f}}_k$ for each block of $\tilde{\boldsymbol{f}}$.

ii). Solve the n tridiagonal systems (2.4.51).

iii). Reassemble these solutions into the blocks $\boldsymbol{p}_k$.

iv). Compute $\tilde{\boldsymbol{u}}_k = S\boldsymbol{p}_k$.

The multiplications by S and S^{-1} can be performed very quickly using the Fast Sine Transform, which is closely related to the FFT. This was originally developed in 1965 by Cooley and Tukey [17], and FFT algorithms are now widely available. In our experiments we used the built-in "FFT" and "IFFT" functions in Matlab.

If we define the $(2n+2) \times 1$ vector $\hat{\boldsymbol{v}}$ by

$$\hat{\boldsymbol{v}} = (0, v_1, \ldots, v_n, 0, \ldots, 0)^T,$$

we then have that

$$\Im m(\mathrm{FFT}(\hat{\boldsymbol{v}})) = (0, -S\boldsymbol{v}, 0, S\boldsymbol{v})^T$$

and

$$4 * \Im m(\mathrm{IFFT}(\hat{\boldsymbol{v}})) = \left(0, S^{-1}\boldsymbol{v}, 0, -S^{-1}\boldsymbol{v}\right)^T,$$

where $\boldsymbol{v} = (v_1, \ldots, v_n)^T$. These algorithms are very fast provided the vector $\hat{\boldsymbol{v}}$ is of length 2^k, $k = 1, 2, \ldots$. For this we must take $n = 2^k - 1$, and provided we do this, the overall operation cost for solving the system (2.4.45) is $\mathcal{O}(n^2 \log n)$. This will be demonstrated by the numerical experiments of §2.6.

2.5 Finding the irregular points

One problem in the implementation of our method which we have not discussed so far is that of determining which mesh points are regular and which are irregular. In [60] no reference is made to this difficulty. In [58], it is claimed that a method exists for determining the irregular points, and this method has an operation cost of only $\mathcal{O}(n)$. However, there is no description of the method at all, and all of the numerical experiments in [58] are carried out on either circles or ellipses, shapes for which it is relatively easy to find the irregular points.

In this section we describe three different algorithms for computing the irregular points, the choice of algorithm depending on the properties of the domain Ω. We shall show that under certain (quite strict) conditions on Ω, namely that $\Omega \in \mathcal{A}$ the set of domains defined below, we have developed an algorithm with an operation cost of $\mathcal{O}(n \log n)$. We have also developed two alternative algorithms, one for Ω convex but $\Omega \notin \mathcal{A}$, and one which will find the irregular points for *any* bounded domain $\Omega \subset \mathbb{R}^2$. Each of these has a cost of $\mathcal{O}(n^2)$ floating point operations (flops), but the algorithm for convex domains is considerably cheaper in terms of the actual number of flops required.

These algorithms are stated in §2.5.2–2.5.4. Now we define the set $\mathcal{A}$.

Definition 2.5.1 *We define the set $\mathcal{A}$ to be the set of bounded domains $\Omega \subset \mathbb{R}^2$ such that:-*

i). *Ω is convex.*

ii). *We can place orthogonal axes on Ω in such a way that if Γ, the boundary of Ω crosses the axes at the points $(x_1, 0)$, $(x_2, 0)$, $(0, y_1)$ and $(0, y_1)$, then Ω is entirely contained within the box with boundaries $y = y_1$, $y = y_2$, $x = x_1$ and $x = x_2$, as shown in Figure 2-3 below.*

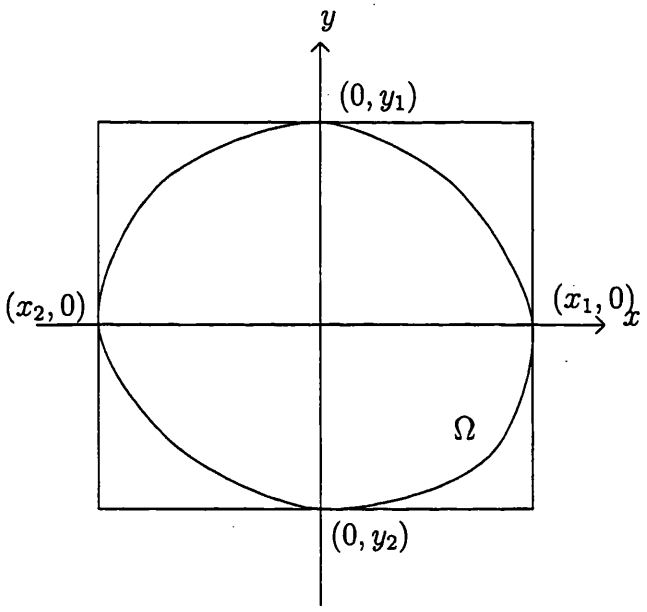

Figure 2-3: A domain $\Omega \in \mathcal{A}$.

Remark: Any axially symmetric convex domain will clearly satisfy (ii), by a suitable choice of axes. Symmetry is not necessary however, as for example any right angled triangle will satisfy (ii), if one axis is taken along the hypoteneuse and the other is taken to pass through the third vertex. An example of a convex domain which does not satisfy (ii) is a non-rectangular parallelogram.

All three algorithms assume that one can determine whether or not a given point lies within a given domain. In certain cases this will be easy to decide,

in other cases it may be harder. It is rather easy if the domain is star-shaped, defined as follows.

Definition 2.5.2 *We define a bounded domain* $\Omega \subset \mathbb{R}^2$ *to be* star shaped *if there exists a point* $\boldsymbol{x} \in \Omega$ *such that one can draw a straight line segment from* $\boldsymbol{x}$ *to every point* $\boldsymbol{y} \in \Gamma$ *in such a way that each line segment lies wholly within the domain* Ω.

Remark: If Ω is a star shaped domain then choosing the origin of coordinates appropriately, the boundary Γ can be written as

$$\Gamma = r(\theta)(\cos(\theta), \sin(\theta)), \quad \theta \in [0, 2\pi], \tag{2.5.52}$$

for some positive valued continuous function r, at least in theory. In practice such a formula may be hard to determine.

It is clear that if we can write the boundary Γ in the form (2.5.52) then the point $\boldsymbol{x} = (x_1, x_2)$ will lie in Ω if

$$|\boldsymbol{x}| < r\left(\tan^{-1}\frac{x_2}{x_1}\right).$$

If this is not the case, or if the boundary is not even given by a known formula, but rather by a series of points which have to be joined in some way, then the problem is more difficult.

In §2.5.1 we describe a general method developed by Gipson [23] for deciding if a point lies inside or outside a given domain. We then describe the three algorithms which can be used for determining the irregular points in §2.5.2–2.5.4. Finally in §2.5.5 we present some numerical results to demonstrate the cost of each method.

2.5.1 Determining if a given point lies within a given domain

The method of [23] is based on the residue theorem from complex analysis. Suppose the function $f(z)$, $z = x + iy$, is meromorphic inside and analytic on a positively oriented contour Γ, except possibly for a finite number of poles at the

points z_k, $k = 1, \ldots, m$, inside Γ. The residue theorem (see for example [70, Theorem 7.4]) then states that

$$\int_\Gamma f(z)dz = 2\pi i \sum_{k=1}^{m} \text{res}\{f(z); z_k\},$$

where $\text{res}\{f(z); z_k\}$ denotes the *residue* of f at z_k. Laurent's Theorem implies that we can write

$$f(z) = \sum_{n=-\infty}^{n=\infty} a_n (z - z_k)^n,$$

in an open disc centred on z_k, and then

$$\text{res}\{f(z); z_k\} = a_{-1}.$$

Thus if $f(z) = 1/(z - z_k)$, then $\text{res}\{f(z); z_k\} = 1$.

Now, suppose we wish to know whether or not a point $z_p \in \mathbb{R}^2$ lies inside or outside a bounded domain $\Omega \subset \mathbb{R}^2$. It shall be assumed here that the boundary Γ of Ω is defined by a discrete set of boundary points, joined by straight line segments. If the points are joined in some way other than straight lines, by spline interpolation for example, then a similar method to that described here can also be applied.

The method works in the following way. First, we place a pole of the form $1/(z - z_p)$ at the test point z_p in the complex plane. We then define the contour integral round the boundary Γ as the sum of the integrals over the line segments Γ_j. Placing the local origin of coordinates at z_p, we can write $z - z_p = re^{i\theta}$, and then we find

$$\int_{\Gamma_j} \frac{dz}{z - z_p} = \int_{\Gamma_j} \frac{dr}{r} + i \int_{\Gamma_j} d\theta. \tag{2.5.53}$$

By the residue theorem, it is clear that only the imaginary part of the contour integral can be non-zero, hence we are only concerned with the imaginary part

of (2.5.53), namely

$$\Im m \int_{\Gamma_j} \frac{dz}{z - z_p} = (\theta_2 - \theta_1),$$

where θ_1 and θ_2 are the angles between the real axis and the endpoints of the boundary segment, as shown in Figure 2-4 below. Hence there is no need to

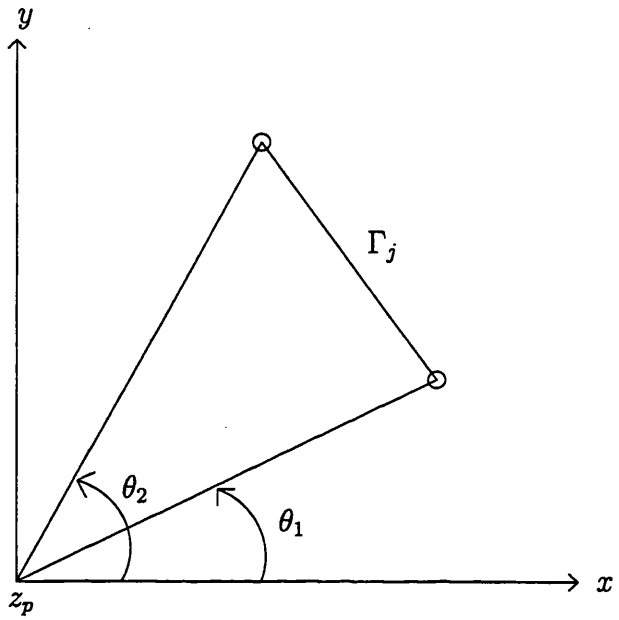

Figure 2-4: Integration over Γ_j.

perform an explicit boundary integration, all that we need to do is to compute the angle $\theta_2 - \theta_1$ for each boundary segment. After integrating over each segment the sum of these angles will either be 2π, in which case we conclude that the point is inside the domain, or zero, in which case we conclude that the point is outside the domain. If the point is on the boundary, then the sum will be somewhere between 0 and 2π, π if the point is a smooth boundary point. For details see [23].

Note that if Γ is given by some explicit functional definition, then this method only determines whether or not z_p lies inside the polygon approximating Γ, and not Γ itself.

The method can also deal with a multiply connected region. To do this the boundary nodes for the external boundary (or boundaries) must be labelled in an anticlockwise direction, and for an internal boundary they must be labelled

in a clockwise direction, as shown in Figure 2-5 below. At the external test point

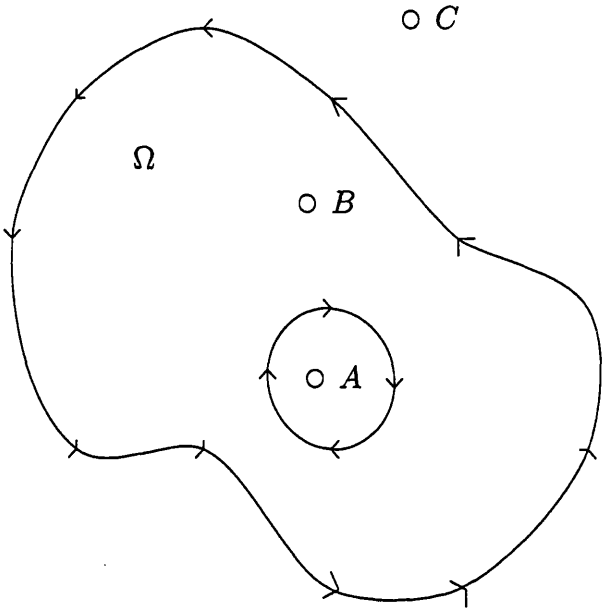

Figure 2-5: Various test points for a multiply connected domain Ω.

A, it is clear that the sum of angles going round the outer boundary will be 2π, but this will be cancelled by the sum going round the inner boundary, which will be -2π. Hence the total sum will be zero, and we conclude that the point lies outside Ω.

At the internal point B, the sum of angles round the outer boundary will be 2π, but the sum round the inner boundary will be zero, as the positive contributions to the sum moving round the inner boundary will be cancelled by the negative contributions. Hence the total sum is 2π, and we conclude that the point lies within Ω.

Finally at the external point C, the sum of angles round both the inner and outer boundaries will be zero, as for each boundary the positive and negative contributions to the sum will cancel each other out. So, the total sum is zero, and we conclude that the point lies outside the domain.

Armed with this method, we can now describe some algorithms for determining the irregular points of Ω.

2.5.2 An algorithm for determining the irregular points for a domain $\Omega \in \mathcal{A}$

If $\Omega \in \mathcal{A}$, then by suitably orientating it before covering it in the mesh R_h (2.3.13) we can ensure that for any row or column of the mesh, no point of that row or column will lie inside the domain unless the centre point (lying on either the x or y axis) does also. Also, as $\Omega \in \mathcal{A}$ implies that Ω is convex, we know that if we move along any row or column from point to point, we will never leave and then re-enter Ω. We can exploit these two facts to design an algorithm which will find the irregular points at a cost of $\mathcal{O}(n \log n)$ floating point operations. This is because we only need to ask the question "is a point inside Ω?" at $\mathcal{O}(n \log n)$ points, as we shall explain below. If the boundary of Γ is given in the form (2.5.52) then this question can be answered cheaply, if not then the question can be answered at a greater cost using the method of §2.5.1. Either way, the order of the cost in terms of n will be the same.

First, recall that n is the number of mesh points in each direction, and because of the constraints of the FFT method of §2.4, we will always have

$$n = 2^k - 1, \quad k = 1, 2, \ldots . \tag{2.5.54}$$

An irregular point is one whose neighbour in one (or more) of the coordinate directions lies the other side of Γ from the point itself. Hence each irregular point in Ω will be next to at least one irregular point in $R \setminus \Omega$, where R is the square (2.3.10). So, a point $(x_i, y_j) \in \Omega$ will be irregular if either $(x_{i-1}, y_j) \notin \Omega$, $(x_{i+1}, y_j) \notin \Omega$, $(x_i, y_{j-1}) \notin \Omega$ or $(x_i, y_{j+1}) \notin \Omega$.

To find all of the irregular points $(x_i, y_j) \in \Omega$ whose neighbour $(x_{i-1}, y_j) \notin \Omega$ (and hence (x_{i-1}, y_j) is an irregular point in $R \setminus \Omega$), we use Algorithm 1, which we explain below. Here k is related to n as in (2.5.54), and the points (x_i, y_j) are given by (2.3.11).

This algorithm works in the following way. For each row (ie. for $j = 1, \ldots, n$), we first consider the central point of the row, $(x_{2^{k-1}}, y_j)$. As $\Omega \in \mathcal{A}$, if any point of the row lies within Ω, then the centre point certainly will. So, if it doesn't, move onto the next row, otherwise continue. Now consider the point halfway between the centre and the left, $(x_{2^{k-2}}, y_j)$. Is this point in Ω? If so, consider the point halfway between it and the left most point of the mesh, $(x_{2^{k-3}}, y_j)$,

Algorithm 1

```
for j = 1, ..., n
    if (x_{2^{k-1}}, y_j) ∈ Ω
      i = 2^{k-2}
      for m = 1, ..., k - 2
          if (x_i, y_j) ∈ Ω
            i = i - 2^{k-2-m}
          else
            i = i + 2^{k-2-m}
          end
      end
      if  (x_i, y_j) ∈ Ω
          (x_i, y_j) is an internal irregular point
          (x_{i-1}, y_j) is an external irregular point
      else
          (x_i, y_j) is an external irregular point
          (x_{i+1}, y_j) is an internal irregular point
      end
    end
end
```

otherwise move halfway back towards the centre, to the point $(x_{3*2^{k-3}}, y_j)$. As we are assuming that the domain is convex, repeating this procedure will eventually lead us to a point whose neighbour will lie the other side of the mesh from the point itself. If the point we are at is inside Ω then it is an internal irregular point, otherwise it is an external irregular point.

In a similar way we can find $(x_i, y_j) \in \Omega$ whose neighbour $(x_{i+1}, y_j) \notin \Omega$, and we can perform a similar procedure on all of the columns to find the other irregular mesh points.

Now consider the cost of this algorithm. There are n rows and n columns, and so as we have to move along each row and each column twice, the cost will be $4n$ times the number of operations on each row or column. One can see from the algorithm that this cost will be $\mathcal{O}(k)$, and from (2.5.54), it is clear that $k = \log_2(n+1)$. Hence the total cost is $\mathcal{O}(n \log n)$. Numerical results demonstrating this will be presented in §2.5.5.

2.5.3 An algorithm for determining the irregular points for a convex domain $\Omega \notin \mathcal{A}$

This algorithm is closely related to Algorithm 1. As Ω is convex, we can pass along each row or column, and it is clear that if we pass from Ω to $R \setminus \Omega$, then we will not re-enter Ω if we keep moving along the row/column in the same direction. The main difference is that we can no longer guarantee that if any point of a row or column lies in Ω then the central point will. Hence we cannot start each loop at the centre of the row/column and use an $\mathcal{O}(k)$ algorithm for finding the boundary. Instead we move along each row and column in turn, starting from the edge of R, until we reach a point which lies inside Ω, and then we know that this point is an irregular point. For example, to find all of the irregular points $(x_i, y_j) \in \Omega$ whose neighbour $(x_{i-1}, y_j) \notin \Omega$ we use Algorithm 2 below.

Algorithm 2

for $j = 1, \ldots, n$
  $i = 1$
  if $(x_i, y_j) \notin \Omega$
    $i = i + 1$
    if $i = n$
      no points of the row lie in Ω
      so move on to the next row
    end
  end
  (x_i, y_j) is an internal irregular point
  (x_{i-1}, y_j) is an external irregular point
end

Similar algorithms can be used to move the other way along each row, and up and down each column, to find the other irregular points. Using this algorithm one has to ask the question "does this point lie inside or outside Ω?" at all mesh points in the grid, except for the regular points inside Ω. Hence the operation cost will be less than if this question was asked at every point of the grid, but it will still be $\mathcal{O}(n^2)$.

2.5.4 A method for determining the irregular points for a non-convex domain Ω

In this section we present a general method which can be used to find the irregular points for any bounded domain $\Omega \subset \mathbb{R}^2$. In theory, as there are $\mathcal{O}(n)$ of these points, and for a given boundary they are all connected to each other, it should be possible to find an $\mathcal{O}(n)$ algorithm for finding them, by moving from irregular point to irregular point somehow. However, in practice, we have not been able to implement a robust algorithm of this form, as it is hard to move from irregular point to irregular point without missing some on the way.

Instead, we simply move through the mesh, and for every point we determine whether or not it lies within Ω. This gives us an n by n matrix, whose entries are one if the point lies inside Ω and zero otherwise. To find the irregular points, one can then just look at each matrix entry, and see if any of its four nearest neighbours are different. If so the point is irregular.

This can be done in a loop, but alternatively, unlike Algorithms 1–2, this method can be implemented using simple matrix operations only. When using Matlab (as we do in our experiments) this drastically reduces the computing time, as demonstrated by the numerical results in §2.5.5.

The cost of this method is $\mathcal{O}(n^2)$, as we have to perform an operation on every mesh point. However, this cost is the same order as the cost of the other components of finding a particular solution (namely the fast solve described in §2.4), and so the advantage to be gained in seeking to reduce it would be minimal. Using the method, we can evaluate the irregular points for complicated, multiply connected regions, such as the "boomerang with a hole in" shape in Figure 2-6 below.

The boundary of the boomerang is given by $\Gamma_1 = (\cos t + 0.65\cos 2t - 0.65, 1.5\sin t)$, and the boundary of the hole is given by $\Gamma_2 = 0.3(\cos t, \sin t)$. This domain is clearly not star shaped, hence we have to use the method of §2.5.1 to determine whether each point lies within the domain or not, and thus we have to approximate the boundary using a suitable polygon.

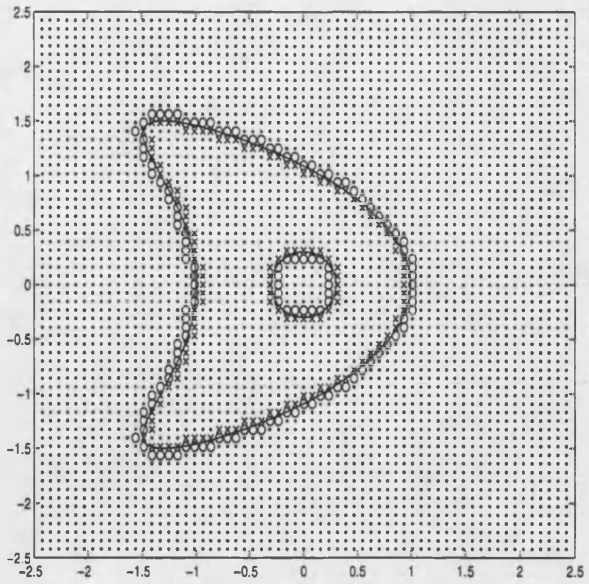

Figure 2-6: Irregular points inside (x) and outside (o) a holed boomerang.

2.5.5 Numerical results

In this section we shall briefly present some numerical results demonstrating the cost of using Algorithms 1 and 2, and also the method of §2.5.4. All of our experiments were carried out using the MATLAB package on a Sun Ultra 2. So that the results could be easily compared, we used each method on an identical shape, the ellipse with boundary $\Gamma = (3\cos t, \sin t)$, $t \in [0, 2\pi]$. We embedded this domain in a square of side length 10.

The operation cost of each method is calulated to be $\mathcal{O}(n^p)$, with p given in the tables. In Table 2.1 we show the number of floating point operations (flops) for each method. In Table 2.2 we show the computing time in seconds (cputime) for each method. In both these tables we merely use the known formula for the boundary of the ellipse to determine whether or not each point lies within the domain. In order to compare the cost of this with the cost of using the method in §2.5.1, in Table 2.3 we give results for the same shape using the methods of §2.5.4 and §2.5.1 together.

The order of the operation cost for each method can be clearly seen, and matches that predicted before. The flop count is clearly least for Algorithm 1. Although Algorithm 2 and the method of §2.5.4 have the same order cost, Algo-

	Algorithm 1		Algorithm 2		Method of §2.5.4	
n	flops	p	flops	p	flops	p
7	764	1.2	1779	1.9	2262	2.2
15	1786	1.3	6669	1.9	10358	2.1
31	4430	1.2	25427	2.0	44214	2.0
63	10222	1.3	100147	2.0	182582	2.0
127	22390	1.1	398693	2.0	741942	2.0
255	49418	1.1	1587553	2.0	2991158	2.0
511	108782		6327731		12011574	

Table 2.1: Flop count for evaluating the irregular points.

	Algorithm 1		Algorithm 2		Method of §2.5.4	
n	cputime	p	cputime	p	cputime	p
7	0.01	1.0	0.02	1.3	0.00	
15	0.02	1.3	0.05	2.1	0.00	
31	0.05	0.8	0.22	2.0	0.01	1.6
63	0.09	1.4	0.87	2.0	0.03	2.3
127	0.23	1.1	3.45	2.0	0.15	2.2
255	0.49	1.2	13.84	1.9	0.70	2.1
511	1.15		52.96		3.01	

Table 2.2: Computer time for evaluating the irregular points.

rithm 2 has roughly half the flop count.

As we are using Matlab, loops are particularly expensive in terms of cpu time, whilst matrix multiplications are much cheaper. Hence although the method of §2.5.4 has the greatest flop count, it does in fact run much faster than Algorithm 2 for all values of n, and it is even quicker than Algorithm 1 up to about $n = 127$.

From Table 2.3 it is clear that using the method of §2.5.1 to decide whether or not a point lies within the domain is much more expensive than using a known formula for a star shaped domain. The cost increases depending on the fineness of the polygon used to approximate Γ. In Table 2.3 the boundary nodes were spaced at intervals of $\pi/20$. Although no further results are shown here, it was found that reducing the spacing of the boundary nodes drastically increased the cost of the method, although for a fixed node spacing the $\mathcal{O}(n^2)$ cost with respect to n can still be clearly seen.

n	flops	p	cputime	p
7	49126	2.2	0.68	2.2
15	224950	2.1	3.14	2.1
31	960214	2.0	13.34	2.1
63	3965206		55.92	

Table 2.3: Using the methods of §2.5.4 and §2.5.1.

2.6 Numerical experiments

In this section we present some numerical results demonstrating the accuracy and cost of the domain embedding method of §2.3–2.5. We used the method to find a particular solution of the inhomogeneous Helmholtz equation

$$-\Delta u(\boldsymbol{x}) + \alpha^2 u(\boldsymbol{x}) = (1+\alpha^2) J_0(|\boldsymbol{x}|), \quad \boldsymbol{x} \in \Omega, \tag{2.6.55}$$

where Ω is the ellipse with boundary $\gamma(t) = (3\cos t, \sin t)$, $t \in [0, 2\pi]$, and J_0 is the Bessel function of first kind of order zero.

To find a particular solution, we embedded Ω in a square of side length 10, covered the square in the mesh (2.3.11), and then solved the extended problem (2.3.20) with boundary conditions given by (2.3.21). The exact value of this particular solution is unknown, so we computed a "near exact" solution by applying the method with a large value of n ($n = 1023$), and then for the purpose of computing errors we took this to be the true solution.

In order to demonstrate that the function which our method converges to really is a particular solution of (2.6.55), in Chapter 4 we combine the methods of Chapters 2–4 to solve the inhomogeneous boundary value problem (2.6.55) with boundary data

$$u(\boldsymbol{x}) = J_0(|\boldsymbol{x}|), \quad \text{on } \Gamma, \tag{2.6.56}$$

as for this problem the exact solution is known, and is given by $u(\boldsymbol{x}) = J_0(|\boldsymbol{x}|)$ (see §4.4 for details).

We solved (2.6.55) for $\alpha = 1$ and for $\alpha = 100$ in order to demonstrate the good performance of the method regardless of the value of α. For each example we computed the approximate solution for $n = 2^k - 1$, $k = 5, \ldots, 9$. We computed

the errors at the points $(0,0)$, $(1.25,0)$ and $(2.5,0)$, and at each point we computed the estimated order of convergence (EOC) to be

$$\text{EOC} = \log_2\left(\frac{\text{error}_{n=2^{k+1}-1}}{\text{error}_{n=2^k-1}}\right). \tag{2.6.57}$$

Using the error bound (2.4.48) we would expect EOC ≈ -2. The results are given in Tables 2.4 and 2.5, and the expected order of convergence is clearly acheived.

n	error(0,0)	EOC	error(1.25,0)	EOC	error(2.5,0)	EOC
31	2.563e-3	-2.0	1.847e-3	-2.1	3.212e-4	-1.6
63	6.229e-4	-2.0	4.228e-4	-2.1	1.094e-4	-1.7
127	1.595e-4	-2.1	1.007e-4	-2.2	3.448e-5	-2.5
255	3.603e-5	-2.3	2.249e-5	-2.3	6.308e-6	-2.2
511	7.175e-6		4.449e-6		1.329e-6	

Table 2.4: Errors for solving (2.6.55) with $\alpha = 1$ by domain embedding.

n	error(0,0)	EOC	error(1.25,0)	EOC	error(2.5,0)	EOC
31	6.075e-7	-2.0	3.890e-7	-2.0	2.414e-5	-12.9
63	1.519e-7	-2.0	9.839e-8	-2.0	3.082e-9	-1.8
127	3.754e-8	-2.1	2.432e-8	-2.1	8.923e-10	-2.1
255	8.939e-9	-2.3	5.792e-9	-2.3	2.127e-10	-2.3
511	1.788e-9		1.158e-9		4.253e-11	

Table 2.5: Errors for solving (2.6.55) with $\alpha = 100$ by domain embedding.

In Table 2.6 we show the operation cost of the method. Assuming the operation cost is $\mathcal{O}(n^p)$, p can be calculated using the formula

$$p = \log_2\left(\frac{\text{flops}_{n=2^{k+1}-1}}{\text{flops}_{n=2^k-1}}\right). \tag{2.6.58}$$

The cost can be broken down into three parts.

i). The cost of computing the irregular points. This has been dealt with in §2.5.

ii). The cost of computing the correction term $c_{i,j}$ (2.3.31) at the irregular points. There are $\mathcal{O}(n)$ irregular points, and for each one the correction

n	flops for computing $c_{i,j}$	p	flops for solving (2.4.45)–(2.4.46)	p
31	18620	1.0	115925	2.1
63	38168	1.0	502499	2.1
127	75812	1.0	2164817	2.1
255	152512	1.0	9292175	2.1
511	307488		39746893	

Table 2.6: Operation cost for solving (2.6.55) by domain embedding.

term is computed by solving the 3×3 system (2.3.43) and the 4×4 system (2.3.44). Thus for each irregular point the cost is independent of n and so we would expect the total cost of computing $c_{i,j}$ to be $\mathcal{O}(n)$.

iii). The cost of solving the linear system (2.4.45)–(2.4.46). As explained in §2.4 we would expect this to be $\mathcal{O}(n^2 \log n)$.

The expected operation costs can clearly be seen in Table 2.6.

As mentioned in §2.3.2, it is claimed in [62] that because the number of irregular points is small compared to the total number of grid points, it is sufficient to only approximate the five-point Laplacian to $\mathcal{O}(h)$ accuracy at the irregular points in order to approximate u to $\mathcal{O}(h^2)$ accuracy. To test this hypothesis we repeated the experiment of Table 2.4, but this time we only computed $c_{i,j}$ (2.3.31) up to the $\mathcal{O}(1)$ term, ie. we did not include the jumps in the third order derivatives of u. The results can be seen in Table 2.7.

n	error(0,0)	EOC	error(1.25,0)	EOC	error(2.5,0)	EOC
31	2.373e-4	-0.9	2.265e-4	-3.1	1.008e-3	-1.0
63	1.247e-4	-1.4	2.646e-5	-1.2	4.906e-4	-3.8
127	4.589e-5	-2.5	1.183e-5	-5.3	3.454e-5	-1.6
255	8.394e-6	-2.1	2.949e-7	2.8	1.125e-5	-1.8
511	2.010e-6		2.089e-6		3.178e-6	

Table 2.7: Errors for solving (2.6.55) with $\alpha = 1$, using only first order corrections at the irregular points.

The convergence is now very erratic, although the errors are generally comparable in magnitude with those acheived in Table 2.4. When applying the domain

embedding method to the solution of parabolic problems (see Chapter 6), we computed $c_{i,j}$ exactly as in (2.3.31), ie. with the higher order correction.

Chapter 3

Solving the homogeneous Helmholtz problem

3.1 Introduction

In this chapter we discuss the boundary integral equation method for the homogeneous positive definite problem

$$-\Delta U + \alpha^2 U = 0 \tag{3.1.1}$$

on a bounded domain $\Omega \subset \mathbb{R}^2$, or on $\Omega' = \mathbb{R}^2 \setminus \bar{\Omega}$, subject to Dirichlet or Neumann conditions on Γ, the boundary of Ω. This is the Helmholtz equation

$$\Delta U + \omega^2 U = 0$$

with purely imaginary wavenumber $\omega = i\alpha$. For large α the problem (3.1.1) is singularly perturbed, and boundary layers will arise in U if the imposed boundary condition is not compatible with the solution $U = 0$ of the reduced problem ((3.1.1) with $\alpha = \infty$).

Boundary integral reformulations of (3.1.1) are well-known. We define the single and double layer potentials by

$$\mathcal{L}_\alpha v(\boldsymbol{x}) = \int_\Gamma \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y}) v(\boldsymbol{y}) d\boldsymbol{y} \tag{3.1.2}$$

and

$$\mathcal{K}_\alpha v(\boldsymbol{x}) = \int_\Gamma \frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial n(\boldsymbol{y})} v(\boldsymbol{y}) d\boldsymbol{y} \tag{3.1.3}$$

respectively, where $\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})$ is the fundamental solution of (3.1.1), which is given by (2.1.5), and $\partial/\partial n(\boldsymbol{y})$ is the derivative with respect to the unit outward normal at $\boldsymbol{y} \in \Gamma$.

Then, for Γ sufficiently smooth, Green's Third Identity asserts that for any solution of (3.1.1) we have

$$\mathcal{K}_\alpha U(\boldsymbol{x}) - \mathcal{L}_\alpha \frac{\partial U}{\partial n}(\boldsymbol{x}) = \begin{cases} -U(\boldsymbol{x}), & \boldsymbol{x} \in \Omega \\ -\frac{1}{2}U(\boldsymbol{x}), & \boldsymbol{x} \in \Gamma \\ 0, & \boldsymbol{x} \in \mathbb{R}^2 \setminus (\Omega \cup \Gamma). \end{cases} \tag{3.1.4}$$

In this chapter we shall be primarily concerned with the second kind boundary integral equations arising from (3.1.1). These are of the form

$$\lambda u + \mathcal{K}_\alpha u = f, \quad \text{on } \Gamma, \tag{3.1.5}$$

with f a given function on Γ and $\lambda \neq 0$ given. This equation arises when (3.1.1) is to be solved in Ω, subject to Neumann data $\partial U/\partial n = g$ on Γ, in which case, using (3.1.4), we have to solve (3.1.5) with $\lambda = 1/2$, $u = U|_\Gamma$ and $f = \mathcal{L}_\alpha g$. An analogous equation arises from the exterior Neumann problem for (3.1.1) but with $\lambda = -1/2$. On the other hand, interior and exterior Dirichlet problems for (3.1.1) can also be solved using integral equations of the form (3.1.5). To do this we use the "indirect" representation $U = \mathcal{K}_\alpha u$ on Ω, and then employ the standard jump relations for the double layer potential to obtain (3.1.5) where now f represents the Dirichlet data, and $\lambda = -1/2$ (for the interior problem) or $\lambda = 1/2$ (for the exterior problem). This background is all well-known, see for example [49]. We shall briefly consider the first kind boundary integral equations arising from (3.1.1) at the end of this chapter, in §3.8.

Here, we are concerned with the formulation of methods for (3.1.5) which work well for α large, as described in Chapter 1. In order to focus attention on the difficulties which arise when α is large, we study problem (3.1.5) in the simplest geometric context where the contour Γ is smooth. We emphasise that

our ideas could also be used when Γ is non-smooth, but additional considerations for dealing with corner singularities would also be necessary, see for example [31] or [50]. We avoid these here.

When Γ is smooth it is natural to solve (3.1.5) by a global approximation technique. Such methods are able to exploit the smoothness of the solution, yielding superalgebraic or even exponential convergence rates.

Throughout this chapter then, we assume that Γ is parametrised by a 2π-periodic function $\gamma : \mathbb{R} \mapsto \Gamma$ which satisfies (1.1.5)–(1.1.7). Equation (3.1.5) can then be written

$$\lambda u(t) + K_\alpha u(t) = f(t), \quad t \in [0, 2\pi], \tag{3.1.6}$$

where (with a slight abuse of notation) u and f now denote $u \circ \gamma$ and $f \circ \gamma$ respectively, and

$$K_\alpha u(t) = \int_0^{2\pi} k_\alpha(t,\tau) u(\tau) d\tau, \tag{3.1.7}$$

where

$$\begin{aligned} k_\alpha(t,\tau) &= \left\{ \left. \frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial n(\boldsymbol{y})} \right|_{\boldsymbol{x}=\gamma(t),\boldsymbol{y}=\gamma(\tau)} \right\} |\gamma'(\tau)| \\ &= \frac{1}{4}\alpha\,|\gamma(t)-\gamma(\tau)|\, H_1^{(1)}(i\alpha\,|\gamma(t)-\gamma(\tau)|) \left[\frac{(\gamma(\tau)-\gamma(t)).\hat{\boldsymbol{n}}(\gamma(\tau))}{|\gamma(t)-\gamma(\tau)|^2} \right] |\gamma'(\tau)|, \end{aligned} \tag{3.1.8}$$

and where $H_1^{(1)}$ is the Hankel function of the first kind of order one, and $\hat{\boldsymbol{n}}$ is the unit outward normal vector to Γ.

We now introduce a class of collocation methods in the $2n$-dimensional space

$$T_n := \operatorname{span}\{\phi_k : k = 0, \ldots, 2n-1\}, \tag{3.1.9}$$

where

$$\phi_k(t) := \begin{cases} \cos kt, & k = 0, \ldots, n \\ \sin(k-n)t, & k = n+1, \ldots, 2n-1, \end{cases} \tag{3.1.10}$$

ie. $T_n = \text{span}\{1, \cos t, \ldots, \cos nt, \sin t, \ldots, \sin(n-1)t\}$. The collocation points will be the equally spaced points in $[0, 2\pi]$

$$t_j^{(n)} = \frac{j\pi}{n}, \quad j = 0, \ldots, 2n-1. \tag{3.1.11}$$

For these points, the corresponding Lagrange interpolating functions in T_n are

$$l_k^{(n)}(t) = \frac{1}{2n}\left\{1 + 2\sum_{m=1}^{n-1}\cos(m(t - t_k^{(n)})) + \cos(n(t - t_k^{(n)}))\right\}, \; k = 0, \ldots, 2n-1,$$

which satisfy (see for example [49, p.161])

$$l_k^{(n)}(t_j^{(n)}) = \delta_{jk}, \quad j, k = 0, \ldots, 2n-1.$$

Let $\mathcal{C}$ denote the space of 2π-periodic continuous functions on $\mathbb{R}$. Then the operator $\mathcal{P}_n : \mathcal{C} \mapsto T_n$ defined by

$$\mathcal{P}_n v = \sum_{k=0}^{2n-1} v(t_k^{(n)}) l_k^{(n)} \tag{3.1.12}$$

is the interpolatory projection, ie. $\mathcal{P}_n v \in T_n$ interpolates v at the points $\{t_j^{(n)}\}_{j=0}^{2n-1}$.

The standard collocation method for (3.1.6) then seeks a solution

$$u_n = \sum_{k=0}^{2n-1} a_k^{(n)} \phi_k \in T_n,$$

where the coefficients $\{a_k^{(n)}\}_{k=0}^{2n-1}$ are defined by requiring that

$$(\lambda u_n + K_\alpha u_n)(t_j^{(n)}) = f(t_j^{(n)}), \quad j = 0, \ldots, 2n-1. \tag{3.1.13}$$

Equivalently the $a_k^{(n)}$ are found by solving the linear system

$$\sum_{k=0}^{2n-1}\left\{\lambda\phi_k(t_j^{(n)}) + K_\alpha\phi_k(t_j^{(n)})\right\} a_k^{(n)} = f(t_j^{(n)}), \quad j = 0, \ldots, 2n-1. \tag{3.1.14}$$

This method is only semidiscrete and is not practical until we specify how the

integrals

$$K_\alpha \phi_k(t_j^{(n)}) = \int_0^{2\pi} k_\alpha(t_j^{(n)}, \tau)\phi_k(\tau)d\tau, \quad j,k = 0,\ldots,2n-1 \tag{3.1.15}$$

should be computed. These are weakly singular integrals, in fact (from (3.1.8), (B.0.1) and [1, (9.6.11)]):

$$k_\alpha(t,\tau) = O(|t-\tau|^2 \log|t-\tau|), \quad \text{as } t \to \tau \text{ for fixed } \alpha,$$

and, for fixed α, there is a well-established "product integration" technique for handling this singularity (see Kress [49]). This involves splitting k_α into the sum

$$k_\alpha(t,\tau) = k_{1,\alpha}(t,\tau)\log 4\sin^2\frac{t-\tau}{2} + k_{2,\alpha}(t,\tau) \tag{3.1.16}$$

with $k_{1,\alpha}$ and $k_{2,\alpha}$ smooth and 2π-periodic. Then the operator K_α can be replaced by the quadrature approximation

$$\begin{aligned}(K_{\alpha,m}v)(t) &= \int_0^{2\pi} \mathcal{P}_m\{k_{1,\alpha}(t,\cdot)v(\cdot)\}(\tau)\log 4\sin^2\frac{t-\tau}{2}d\tau \\ &\quad + \int_0^{2\pi} \mathcal{P}_m\{k_{2,\alpha}(t,\cdot)v(\cdot)\}(\tau)d\tau,\end{aligned}$$

for some $m \in \mathbb{N}$ which has to be chosen. In view of (3.1.12) the computation of the right hand side of this expression requires the integrals of each of the basis functions $\phi_k(\tau)$ and their products with $\log 4\sin^2\frac{t-\tau}{2}$ over the domain $[0,2\pi]$, all of which are known analytically [49].

Using this in (3.1.13), we get a discrete collocation method with solution $u_{n,m} \in T_n$ satisfying

$$\lambda u_{n,m}(t_j^{(n)}) + K_{\alpha,m}u_{n,m}(t_j^{(n)}) = f(t_j^{(n)}), \quad j = 0,\ldots,2n-1. \tag{3.1.17}$$

For the special choice $m = n$, the solution $u_{n,n}$ of (3.1.17) is closely related to the Nyström method of [49], which approximates (3.1.6) by

$$\lambda\tilde{u}_{n,n}(t) + K_{\alpha,n}\tilde{u}_{n,n}(t) = f(t) \quad \text{for all } t \in [0,2\pi]. \tag{3.1.18}$$

Since $K_{\alpha,n}\tilde{u}_{n,n} = K_{\alpha,n}\mathcal{P}_n\tilde{u}_{n,n}$, it is easily seen that $\mathcal{P}_n\tilde{u}_{n,n}$ satisfies (3.1.17), and by uniqueness of the solution of (3.1.17) (shown below), we have $\mathcal{P}_n\tilde{u}_{n,n} = u_{n,n}$ and hence $\tilde{u}_{n,n}(t_j^{(n)}) = u_{n,n}(t_j^{(n)})$, for each j.

However, neither (3.1.17) nor (3.1.18) is a good method for large α, since methods based on the splitting (3.1.16) degenerate badly as $\alpha \to \infty$. In fact, $\tilde{u}_{n,n}$ satisfies an error estimate of the form

$$\|u - \tilde{u}_{n,n}\|_{L_2} \leq C(\alpha)\frac{1}{n^p}\|u\|_{H^p} \tag{3.1.19}$$

for all $p > 1/2$, where u is the solution of (3.1.6), and $\|\cdot\|_{H^p}$ is the usual Sobolev norm, defined in Appendix A. For fixed α the convergence is superalgebraic with respect to n. However, $C(\alpha)$ grows exponentially in α as $\alpha \to \infty$. This is because $k_\alpha(t,\tau)$ becomes highly peaked near $t = \tau$, and the functions $k_{1,\alpha}(t,\tau)$ and $k_{2,\alpha}(t,\tau)$ in (3.1.16) have opposite signs but each blow up in modulus like $\alpha|\gamma(t) - \gamma(\tau)|I_1(\alpha|\gamma(t) - \gamma(\tau)|)$ away from $t = \tau$, where $I_1(z)$ is the modified Bessel function of order one, which grows exponentially in z.

The following numerical example demonstrates how the method can break down when α is large. We solve (3.1.6), with $\lambda = -1/2$ and $f(t) = 1$, with Γ taken to be the unit circle, using the method (3.1.18). We evaluate $U(\boldsymbol{x}) := \mathcal{K}_\alpha u(\boldsymbol{x})$ (as specified by (3.1.3)) at the point $(0.5, 0)$ for $\alpha = 10$ and $\alpha = 20$. The exact value in each case is $I_0(\alpha/2)/I_0(\alpha)$, which has numerical values $9.7e-3$ for $\alpha = 10$ and $6.5e-5$ for $\alpha = 20$. Table 3.1 shows the error for several values of n. The superalgebraic convergence rate can be clearly seen for $\alpha = 10$, but for $\alpha = 20$ there is no convergence, and the error is of the same magnitude as the true solution. This may be because n is not large enough to demonstrate the asymptotic convergence rate, or else it may be that arithmetic overflow is destroying the convergence of the method. These results were obtained using the MATLAB package on a Sun Ultra 2.

In this chapter we describe a new discrete collocation method which avoids the splitting (3.1.16) and is robust when α is large. This is achieved by using a rescaling technique to transform the integrals into a more benign form. These are then approximated using a variant of the product trapezoidal rule on a suitably graded mesh. The new quadrature rules have fixed rate of convergence $1/m^2$ with respect to the number of quadrature points m (ie. slower than the superalgebraic

n	error($\alpha = 10$)	error($\alpha = 20$)
8	9.7e-3	6.5e-5
16	1.7e-7	6.5e-5
32	2.0e-10	7.4e-5
64		9.1e-5
128		6.1e-5
256		5.7e-5
512		7.2e-5
1024		6.7e-5

Table 3.1: Numerical results for method (3.1.18).

rates which we had, at least in theory, above), but crucially the resulting convergence estimate has an asymptotic constant independent of α. We use these rules to define a new quadrature approximation of $K_\alpha v$. We again denote this by $K_{\alpha,m}v$ and the new discrete collocation solution is again given by (3.1.17). We prove in §3.5 that, for this new method, $u_{n,m}$ satisfies

$$\|u - u_{n,m}\|_{L_2} \leq C \left\{ \frac{1}{n^p} \|u\|_{H^p} + \frac{1}{m^2} \|u\|_{L_2} \right\} \tag{3.1.20}$$

for all $p > 1/2$. Here C is independent of α, n, m and u, ie. the method is robust to large α, and algebraically convergent with respect to m but still superalgebraically convergent with respect to n. Higher order quadrature rules may also be applied to the rescaled integrals, with the result that the $1/m^2$ term in (3.1.20) will be replaced by $1/m^q$ with $q > 2$. To avoid a lengthy exposition we do not give this generalisation here.

Note that up until now we have assumed that the right hand side f of (3.1.6) is known exactly. We shall also need to consider the case where f is a single layer potential of the form (3.1.2), which then must also be computed using some quadrature method, and we do this in §3.7.

The problem of solving (3.1.6) (and its first kind analogue involving the operator $\mathcal{L}_\alpha$), with α large, has not received much attention in the literature. As discussed in Chapter 1, integral equations of this form can arise when boundary integral equation methods are applied to the solution of parabolic PDEs such as the heat equation, and this is the case in both Lubich and Schneider [56]

and Chapko and Kress [12]. However, in both of these papers the problem of constructing a quadrature method for (3.1.6) which is robust as $\alpha \to \infty$ is not tackled, although this is identified as a difficulty in [12].

Integral equations of the form (3.1.6) also appear in Ramesh and Lean [74]. There it is proposed to compute the boundary integrals for large α using series expansions. However, an analysis of this method is not given, and the number of terms to be summed in the expansions will grow rapidly as $\alpha \to \infty$.

The layout of this chapter is as follows. First, in §3.2–3.6 we consider the solution of the second kind equation (3.1.6), where the right hand side f is known exactly. In §3.2 we state and prove some key properties of the double layer operator K_α which will be required in the error analysis. In §3.3 we present an error analysis for the (semi-discrete) collocation method which demonstrates that the rate of convergence of this method is independent of α. In §3.4 we explain how we evaluate the integrals resulting from the collocation method, using a non-standard quadrature scheme, and then in §3.5 we present a fully discrete error analysis for the method, obtaining the estimate (3.1.20). In §3.6 we provide some numerical examples to demonstrate the theoretical properties of the method.

In §3.7 we discuss the quadrature approximation of the single layer potential (3.1.2), and how this affects the solution of the second kind equation (3.1.6) when the right hand side involves a single layer potential. Finally in §3.8 we discuss how the method can also be applied to the solution of the first kind integral equations arising from (3.1.1).

The question of how to compute the solution of (3.1.1) quickly at many points throughout Ω is addressed in Chapter 4, where the overall cost of the method for solving the boundary value problem for (3.1.1) is also discussed.

3.2 Properties of the double layer potential

In this section we shall derive some key properties of the double layer potential (3.1.3), which will be required for the error analysis of our method. Define

$$\begin{aligned} \Psi(x) &= xH_1^{(1)}(ix) \\ r(t,\tau) &= |\gamma(t)-\gamma(\tau)| \\ w(t,\tau) &= \frac{1}{4}\left[\frac{(\gamma(\tau)-\gamma(t)).\hat{\boldsymbol{n}}(\gamma(\tau))}{|\gamma(t)-\gamma(\tau)|^2}\right]|\gamma'(\tau)|, \end{aligned} \tag{3.2.21}$$

and then recalling the parametrised form of the double layer potential (3.1.7), (3.1.8), and using the periodicity of r and w, we can write, for any 2π-periodic function $v : [0, 2\pi] \mapsto \mathbb{R}$,

$$(K_\alpha v)(t) = \int_{t-\pi}^{t+\pi} \Psi(\alpha r(t,\tau))w(t,\tau)v(\tau)d\tau. \tag{3.2.22}$$

Note that

$$w(t,\tau) = \frac{\pi}{2}\frac{\partial}{\partial n(\boldsymbol{y})}\left\{\frac{1}{2\pi}\log|\boldsymbol{x}-\boldsymbol{y}|\right\}\Big|_{\boldsymbol{x}=\gamma(t),\boldsymbol{y}=\gamma(\tau)}.$$

Hence w is a constant multiple of the kernel of the double layer potential arising from Laplace's equation. Under the assumption that γ satisfies (1.1.5)–(1.1.7), it can be shown using a Taylor series argument that

$$w \in C^\infty([0,2\pi]\times[0,2\pi]). \tag{3.2.23}$$

We shall now state and prove some key properties of the functions Ψ and r which we shall require later on. Throughout the rest of this chapter, various constants whose values may vary from line to line, but which are always independent of α, n and m will appear. These are labelled C, C_1, C_2, etc..

Proposition 3.2.1 *There exists a constant C such that*

$$\left|\frac{\partial}{\partial t}r(t,\tau)\right| \le C$$

for all $t, \tau \in [0, 2\pi]$.

Proof: Using elementary calculus we have

$$\left|\frac{\partial}{\partial t}r(t,\tau)\right| = \left|\frac{(\gamma(t)-\gamma(\tau)).\gamma'(t)}{|\gamma(t)-\gamma(\tau)|}\right| \leq |\gamma'(t)|, \tag{3.2.24}$$

for all $t, \tau \in [0, 2\pi]$, and (1.1.5) ensures the required result. □

The next two propositions can be deduced from standard references on Bessel functions, eg. [1].

Proposition 3.2.2 *(i)* $|\Psi(x)|$ *is decreasing on* $[0, \infty)$.
(ii) For all $\rho \in [0, 1)$, *there exists a constant* $C = C(\rho)$ *such that* $|\Psi(x)| \leq Ce^{-\rho x}$, $x \in [0, \infty)$.

Proposition 3.2.3 *(i) There exists a constant* C *such that* $|\Psi'(x)|$ *is decreasing on* $[C, \infty)$.
(ii) For all $\rho \in (0, 1)$, *there exists a constant* $C = C(\rho)$ *such that* $|\Psi'(x)| \leq Ce^{-\rho x}$, $x \in [0, \infty)$.

Proposition 3.2.4 *For all* $t \in [0, 2\pi]$, $\tau \in [t - \pi, t + \pi]$, *there exist constants* $C_1, C_2 > 0$ *such that*

$$C_1|t - \tau| \leq r(t, \tau) \leq C_2|t - \tau|. \tag{3.2.25}$$

Proof: The right hand inequality follows immediately from (3.2.21) and (1.1.5). To prove the left hand inequality, set $\Upsilon(t, \tau) = r(t, \tau)/|t - \tau|$, $t \neq \tau$. Then by (1.1.5), $\lim_{t \to \tau} \Upsilon(t, \tau) = |\gamma'(t)| > 0$, and so $\Upsilon(t, \tau)$ is positive valued everywhere on the compact domain $[0, 2\pi] \times [-\pi, 3\pi]$, and hence the result. □

Before we can establish some bounds on the operator K_α we need the following classical result, see [71, p.86] and the references therein.

Theorem 3.2.5 *Let* $\theta \in (\Theta\phi)(t) := \int_{-\infty}^{\infty} \theta(t - s)\phi(s)ds \leq p \leq \infty$, *the operator*

$$(\Theta\phi)(t) := \int_{-\infty}^{\infty} \theta(t - s)\phi(s)ds \tag{3.2.26}$$

is bounded on $L_p(\mathbb{R})$ *and*

$$\|\Theta\|_{L_p(\mathbb{R})} \leq \|\theta\|_{L_1(\mathbb{R})}.$$

Proof: For a full proof of this theorem we refer to [71]. □

We shall now use these results to establish some bounds on the operator K_α defined by (3.1.7).

Lemma 3.2.6 *There exists a constant C such that, for all $\alpha \geq 1$,*

$$\|K_\alpha\|_{L_2} \leq \frac{C}{\alpha}.$$

Proof: Recall that $\|\cdot\|_{L_2}$ denotes the norm over $[0, 2\pi]$. Using (3.2.22), (3.2.23), (3.2.25), and Propositions 3.2.2(ii) and 3.2.4 we have

$$\begin{aligned} |(K_\alpha v(t))| &\leq C\int_{t-\pi}^{t+\pi} |\Psi(\alpha r(t,\tau))||v(\tau)|d\tau \leq C\int_{t-\pi}^{t+\pi} e^{-\rho\alpha r(t,\tau)}|v(\tau)|d\tau \\ &\leq C\int_{t-\pi}^{t+\pi} e^{-\rho\alpha C_1|t-\tau|}|v(\tau)|d\tau. \end{aligned}$$

Since $t \in [0, 2\pi]$ we then have

$$\begin{aligned} |(K_\alpha v)(t)| &\leq C\int_{-\pi}^{3\pi} e^{-\rho\alpha C_1|t-\tau|}|v(\tau)|d\tau \\ &= C\int_{-\infty}^{\infty} e^{-\rho\alpha C_1|t-\tau|}|v_e(\tau)|d\tau = C(\Theta_\alpha(|v_e|))(t), \end{aligned}$$

where the function v_e is defined by

$$v_e(\tau) := \begin{cases} v(\tau), & \tau \in [-\pi, 3\pi] \\ 0 & \tau \in [-\infty, \infty]/[-\pi, 3\pi], \end{cases}$$

and Θ_α is of the form (3.2.26) with $\theta_\alpha(x) = e^{-\rho\alpha C_1|x|}$, $x \in \mathbb{R}$. Thus,

$$\begin{aligned} \|K_\alpha v\|_{L_2} &= \left\{\int_0^{2\pi} |(K_\alpha v)(t)|^2 dt\right\}^{\frac{1}{2}} \leq C\left\{\int_0^{2\pi} |(\Theta_\alpha(|v_e|))(t)|^2 dt\right\}^{\frac{1}{2}} \\ &\leq C\left\{\int_{-\infty}^{\infty} |(\Theta_\alpha(|v_e|))(t)|^2 dt\right\}^{\frac{1}{2}} = C\|(\Theta_\alpha(|v_e|))\|_{L_2(\mathbb{R})} \\ &\leq C\|\theta_\alpha\|_{L_1(\mathbb{R})}\|v_e\|_{L_2(\mathbb{R})} \leq C\int_{-\infty}^{\infty} e^{-\rho\alpha C_1|x|}dx\|v_e\|_{L_2(\mathbb{R})}, \end{aligned}$$

where the last inequality follows from Theorem 3.2.5, and then using the period-

icity of v,

$$\|K_\alpha v\|_{L_2} \leq \sqrt{2}C\int_0^\infty e^{-\rho\alpha C_1 x}dx\|v\|_{L_2} \leq \sqrt{2}C\frac{1}{\rho\alpha C_1}\|v\|_{L_2}$$

and hence the result. □

Lemma 3.2.7 *There exists a constant C such that for all $\alpha \geq 1$,*

$$\|K_\alpha\|_{L_2\to H^1} \leq C.$$

Proof: First note that, for $v \in H^1$, $\|v\|_{H^1}^2 = \|v\|_{L_2}^2 + \|v'\|_{L_2}^2$. Now, by (3.2.22) and standard results on differentiation of functions defined as integrals, it is easy to see that $K_\alpha v \in \mathcal{C}^1(\mathbb{R}) \subset H^1$, and that for $t \in \mathbb{R}$,

$$\begin{aligned}\frac{d}{dt}(K_\alpha v)(t) &= \int_0^{2\pi} \alpha\left\{\frac{\partial}{\partial t}r(t,\tau)\right\}\Psi'(\alpha r(t,\tau))w(t,\tau)v(\tau)d\tau \\ &\quad + \int_0^{2\pi}\Psi(\alpha r(t,\tau))\frac{\partial}{\partial t}w(t,\tau)v(\tau)d\tau \\ &=: I_{1,\alpha}v(t) + I_{2,\alpha}v(t).\end{aligned}$$

Then, by periodicity and (3.2.23) we get

$$|I_{2,\alpha}v(t)| \leq C\int_{t-\pi}^{t+\pi}|\Psi(\alpha r(t,\tau))||v(\tau)|d\tau,$$

and using Proposition 3.2.1 as well, we also get

$$|I_{1,\alpha}v(t)| \leq C\alpha\int_{t-\pi}^{t+\pi}|\Psi'(\alpha r(t,\tau))||v(\tau)|d\tau.$$

Then recalling Propositions 3.2.2(ii) and 3.2.3(ii), and using exactly the same argument as for Lemma 3.2.6 we get:-

$$\|I_{2,\alpha}v\|_{L_2} \leq \frac{C}{\alpha}\|v\|_{L_2}$$

and

$$\|I_{1,\alpha}v\|_{L_2} \leq C\|v\|_{L_2}.$$

Hence, for all $\alpha \geq 1$

$$\left\| \frac{d}{dt} K_\alpha v \right\|_{L_2} \leq C \|v\|_{L_2}$$

and hence the result. □

We remark that by operator interpolation theory (see for example [8, Chapter 12] or [72, p.32]), Lemmas 3.2.6 and 3.2.7, can be generalised to obtain

$$\|K_\alpha\|_{L_2 \to H^\lambda} \leq \|K_\alpha\|^{1-\lambda}_{L_2 \to L_2} \|K_\alpha\|^{\lambda}_{L_2 \to H^1} \leq C \left(\frac{1}{\alpha}\right)^{1-\lambda},$$

for all $\lambda \in [0,1]$. However this more general result is not needed for the error analysis of the next section.

3.3 Error analysis for the collocation method

In this section we present an error analysis for the collocation method (3.1.13) without numerical integration, and we demonstrate the robustness of its convergence rate with respect to α. This result is then needed in §3.5 to obtain the analogous robustness of the fully discrete method. The main result of this section is as follows:-

Theorem 3.3.1 *Assuming the integrals $K_\alpha \phi_k$ in (3.1.14) are known exactly, then there exists a constant n_0 independent of α such that, for $n \geq n_0$, the collocation equations (3.1.13) have a unique solution $u_n \in T_n$, and we have the error estimate*

$$\|u - u_n\|_{L_2} \leq C \frac{1}{n^p} \|u\|_{H^p}, \quad \text{for all } p > \frac{1}{2},$$

where u is the solution of (3.1.6) and C is independent of α and n.

Proof: Observe that by the definition (3.1.12) of $\mathcal{P}_n$, $u_n \in T_n$ satisfies

$$(\lambda I + \mathcal{P}_n K_\alpha) u_n(t) = \mathcal{P}_n f(t).$$

So, if $(\lambda I + \mathcal{P}_n K_\alpha)^{-1}$ exists, then standard arguments show

$$u - u_n = u - (\lambda I + \mathcal{P}_n K_\alpha)^{-1}\mathcal{P}_n f = \lambda(\lambda I + \mathcal{P}_n K_\alpha)^{-1}(u - \mathcal{P}_n u).$$

From [51, p.202] we have the following estimate for the approximation power of $\mathcal{P}_n$:

$$\|v - \mathcal{P}_n v\|_{H^q} \leq C\frac{1}{n^{p-q}}\|v\|_{H^p}, \quad 0 \leq q \leq p, \quad p > \frac{1}{2}. \tag{3.3.27}$$

Hence

$$\|u - u_n\|_{L_2} \leq C\|(\lambda I + \mathcal{P}_n K_\alpha)^{-1}\|_{L_2}\frac{1}{n^p}\|u\|_{H^p},$$

for all $p > 1/2$, and the result follows provided we show that, for $n \geq n_0$, $(\lambda I + \mathcal{P}_n K_\alpha)^{-1}$ exists, and $\|(\lambda I + \mathcal{P}_n K_\alpha)^{-1}\|_{L_2}$ is bounded by a constant independently of α and n. This result will be proved in Theorem 3.3.10 below. □

Before we prove Theorem 3.3.10, we shall show that for all $\alpha \geq 1$, $(\lambda I + K_\alpha)^{-1}$ exists and is bounded independently of α in $\|\cdot\|_{L_2}$. First we quote the following well known result, which appears as Theorem 5 in [4].

Theorem 3.3.2 *Let X and Y be normed linear spaces, at least one of which is complete. Let $S, T \in \mathcal{L}(X, Y)$ (the linear space of all bounded and linear operators from X to Y), and let $S^{-1} \in \mathcal{L}(Y, X)$. Also assume*

$$\|S - T\| < \frac{1}{\|S^{-1}\|}. \tag{3.3.28}$$

Then T^{-1} exists as a bounded linear operator from Y onto X, and

$$\|T^{-1}\| \leq \frac{\|S^{-1}\|}{1 - \|S^{-1}\|\|S - T\|}. \tag{3.3.29}$$

Proof: See [4]. Note also that if we take $S = I$, $T = I - A$, then this is the same as [49, Theorem 2.8], where an alternative proof for this specific case can be found. □

We can then get the following result straight away.

Lemma 3.3.3 *For α sufficiently large, $(\lambda I + K_\alpha)^{-1}$ exists and $\|(\lambda I + K_\alpha)^{-1}\|_{L_2}$ is bounded independently of α.*

Proof: Recall that, from Lemma 3.2.6, there exists a constant C independent of n and α such that $\|K_\alpha\|_{L_2} \leq C/\alpha$. Hence, for α sufficiently large (specifically $\alpha > C/|\lambda|$), $\|K_\alpha\|_{L_2} < |\lambda|$. Hence by Theorem 3.3.2 (with $S = \lambda I$, $T = \lambda I + K_\alpha$) we have that $(\lambda I + K_\alpha)^{-1}$ exists as a bounded linear operator from L_2 onto L_2, and

$$\|(\lambda I + K_\alpha)^{-1}\|_{L_2} \leq \frac{1}{|\lambda| - \|K_\alpha\|_{L_2}} \leq \frac{1}{|\lambda| - C/\alpha} \leq C',$$

provided $\alpha \geq C/(|\lambda| - (C')^{-1})$. □

This proof requires α to be sufficiently large. We improve this result below and prove the result to be true for all $\alpha \geq 1$, but this requires a little more work. We begin by establishing the following result.

Lemma 3.3.4 *The operator $(\lambda I + K_\alpha)$ (where $\lambda = \pm 1/2$) has no zero eigenvalues.*

Proof: Suppose that $u \in L_2[0, 2\pi]$ and $(\lambda I + K_\alpha)u = 0$. Then, by Lemma 3.2.7, $u \in H^1 \subset C(\mathbb{R})$. Defining the function

$$v(\boldsymbol{x}) = \frac{u(\gamma^{-1}(\boldsymbol{x}))}{|\gamma'(\gamma^{-1}(\boldsymbol{x}))|}, \quad \boldsymbol{x} \in \Gamma,$$

we have that $v \in C(\Gamma)$ and

$$(\lambda I + \mathcal{K}_\alpha)v = 0. \tag{3.3.30}$$

Now (recalling that (3.1.3) defines $\mathcal{K}_\alpha v(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^2$), we define

$$\begin{aligned} U_i(\boldsymbol{x}) &:= \mathcal{K}_\alpha v(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \\ U_o(\boldsymbol{x}) &:= \mathcal{K}_\alpha v(\boldsymbol{x}), \quad \boldsymbol{x} \notin \Omega \cup \Gamma. \end{aligned}$$

Then by the linearity of the integral operator $\mathcal{K}_\alpha$ it is clear that both U_i and U_o satisfy (3.1.1). By the jump relation properties of the double layer potential (see

for example [49, Theorem 6.13]), we have

$$\begin{aligned} U_i|_\Gamma &= \left(-\frac{1}{2}I + \mathcal{K}_\alpha\right) v\Big|_\Gamma \\ U_o|_\Gamma &= \left(\frac{1}{2}I + \mathcal{K}_\alpha\right) v\Big|_\Gamma, \end{aligned} \tag{3.3.31}$$

and so

$$U_i|_\Gamma - U_o|_\Gamma = -v. \tag{3.3.32}$$

Now suppose that $\lambda = -1/2$ (the proof for $\lambda = 1/2$ is similar). Then by (3.3.30) and (3.3.31) $U_i|_\Gamma = 0$, and so U_i satisfies the homogeneous Helmholtz equation inside a domain Ω, with zero Dirichlet data. Hence $U_i(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \Omega$, by the uniqueness of the solution of the interior Dirichlet Helmholtz problem (see for example [15, §3.3]). Thus, with $\boldsymbol{n}$ denoting the unit outward normal vector to Γ, and with

$$\frac{\partial u_\pm}{\partial n}(\boldsymbol{x}) := \lim_{h \to 0} \boldsymbol{n}.\nabla u(\boldsymbol{x} \pm h\boldsymbol{n}(\boldsymbol{x}))$$

to be understood in the sense of uniform convergence on Γ, we have

$$\frac{\partial U_{i-}}{\partial n}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Gamma.$$

As the normal derivative of the double layer potential is continuous across the boundary (see for example [49, Theorem 6.13]), we also have

$$\frac{\partial U_{o+}}{\partial n}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Gamma.$$

Noting that U_0 satisfies the Sommerfeld radiation condition, and appealing to the uniqueness of the solution of the exterior Neumann Helmholtz problem subject to this condition (see for example [15, §3.3]), we must also have $U_o(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \notin \Omega \cup \Gamma$, and so by (3.3.32) we have $v = 0$ and hence the result. □

This is enough to prove existence and boundedness of $(\lambda I + K_\alpha)^{-1}$ for all $\alpha \geq 1$.

Lemma 3.3.5 *For all $\alpha \geq 1$, $(\lambda I + K_\alpha)^{-1}$ exists and is bounded in $\|\cdot\|_{L_2}$.*

Proof: By [49, Theorem 2.22], any integral operator with a continuous or weakly singular kernel is a compact operator on $L_2(\Gamma)$. Hence K_α is compact, and so by Lemma 3.3.4 and the Fredholm Alternative [49, Corollary 3.5], $(\lambda I + K_\alpha)$ must be a bijective bounded linear operator. Hence by Banach's Isomorphism Theorem (see for example [73, p.145]), $(\lambda I + K_\alpha)^{-1}$ exists, and $\|(\lambda I + K_\alpha)^{-1}\|_{L_2} < \infty$. □

We now need to show that the bound on $\|(\lambda I + K_\alpha)^{-1}\|_{L_2}$ is independent of α. We first need the following two results.

Lemma 3.3.6 *For all $\alpha \geq 1$, $\|K_\alpha - K_{\alpha'}\|_{L_2}$ tends to zero as α' tends to α.*

Proof: First note that with Ψ defined as in §3.2 we can write, for $x, \tilde{x} \in \mathbb{R}$,

$$\begin{aligned}\Psi(x) - \Psi(\tilde{x}) &= \int_0^1 \frac{d}{d\theta}\{\Psi(\theta x + (1-\theta)\tilde{x})\}\, d\theta \\ &= (x - \tilde{x}) \int_0^1 \Psi'(\theta x + (1-\theta)\tilde{x}) d\theta. \qquad (3.3.33)\end{aligned}$$

Then recalling the definition of K_α from (3.2.22) we have

$$\begin{aligned}|(K_\alpha - K_{\alpha'})v(t)| &= \left| \int_{t-\pi}^{t+\pi} (\Psi(\alpha r(t,\tau)) - \Psi(\alpha' r(t,\tau))) w(t,\tau) v(\tau) d\tau \right| \\ &= \left| \int_{t-\pi}^{t+\pi} (\alpha - \alpha') r(t,\tau) \int_0^1 \Psi'\left[(\theta\alpha + (1-\theta)\alpha') r(t,\tau)\right] d\theta\, w(t,\tau) v(\tau) d\tau \right| \\ &\leq C|\alpha - \alpha'| \int_0^1 \underbrace{\left\{ \int_{t-\pi}^{t+\pi} |\Psi'((\theta\alpha + (1-\theta)\alpha') r(t,\tau))|^2 d\tau \right\}^{\frac{1}{2}}}_{I} d\theta\, \|v\|_{L_2}\end{aligned}$$

where the last line follows using (3.2.23) and the Cauchy-Schwartz inequality. Using Propositions 3.2.3(ii) and 3.2.4 we then have

$$\begin{aligned}I^2 &\leq C \int_{t-\pi}^{t+\pi} e^{-2\rho(\theta\alpha + (1-\theta)\alpha')C_1|t-\tau|} d\tau \\ &= 2C \int_0^{\pi} e^{-2\rho(\theta\alpha + (1-\theta)\alpha')C_1 x} dx \\ &\leq \frac{C}{(\theta\alpha + (1-\theta)\alpha')}.\end{aligned}$$

Hence

$$\begin{aligned} |(K_\alpha - K_{\alpha'})v(t)| &\leq C|\alpha - \alpha'| \int_0^1 (\theta\alpha + (1-\theta)\alpha')^{-\frac{1}{2}} d\theta \, \|v\|_{L_2} \\ &= C|\alpha - \alpha'| \frac{2}{\sqrt{\alpha} + \sqrt{\alpha'}} \|v\|_{L_2}. \end{aligned} \tag{3.3.34}$$

So, $\|(K_\alpha - K_{\alpha'})v\|_{L_2} \leq C(|\alpha - \alpha'|/(\sqrt{\alpha} + \sqrt{\alpha'}))\|v\|_{L_2}$, and

$$\begin{aligned} \|K_\alpha - K_{\alpha'}\|_{L_2} &\leq C \frac{|\alpha - \alpha'|}{\sqrt{\alpha} + \sqrt{\alpha'}} \\ &\leq C|\alpha - \alpha'| \quad (\text{for } \alpha \geq 1), \end{aligned}$$

and hence the result. □

Lemma 3.3.7 *$\|(\lambda I + K_\alpha)^{-1}\|_{L_2}$ is continuous with respect to α.*

Proof: In this proof we take all norms to be $\|\cdot\|_{L_2}$. First note that by Lemma 3.3.5 we know that $(\lambda I + K_\alpha)^{-1}$ exists. Then using Theorem 3.3.2 with $T = (\lambda I + K_{\alpha'})$ and $S = (\lambda I + K_\alpha)$, we have

$$\begin{aligned} \|S - T\| &= \|K_\alpha - K_{\alpha'}\| \\ &< \frac{1}{(\lambda I + K_\alpha)^{-1}} \end{aligned}$$

for α' sufficiently close to α (by Lemma 3.3.6). So, S and T satisfy the assumptions in Theorem 3.3.2, and so we have

$$\begin{aligned} \|(\lambda I + K_{\alpha'})^{-1}\| &\leq \frac{\|(\lambda I + K_\alpha)^{-1}\|}{1 - \|(\lambda I + K_\alpha)^{-1}\|\|K_{\alpha'} - K_\alpha\|} \\ &\leq 2\|(\lambda I + K_\alpha)^{-1}\| \end{aligned}$$

provided we choose α' close enough to α that

$$\|K_{\alpha'} - K_\alpha\| \leq \frac{1}{2\|(\lambda I + K_\alpha)^{-1}\|}.$$

Then for α and α' sufficiently close we have

$$\begin{aligned} \|(\lambda I + K_{\alpha'})^{-1} - (\lambda I + K_\alpha)^{-1}\| &= \|(\lambda I + K_\alpha)^{-1}(K_\alpha - K_{\alpha'})(\lambda I + K_{\alpha'})^{-1}\| \\ &\leq 2\|(\lambda I + K_\alpha)^{-1}\|^2 \|K_\alpha - K_{\alpha'}\|, \end{aligned}$$

which tends to zero as α' tends to α, and hence the result. □

We can now prove the following lemma.

Lemma 3.3.8 *There exists a constant C independent of α such that $\|(\lambda I + K_\alpha)^{-1}\|_{L_2} \leq C$, for all $\alpha \geq 1$.*

Proof: Note that by Lemma 3.3.3, we have shown the result to be true for α large enough. Hence, there exists a constant A such that for all $\alpha \in [A, \infty)$,

$$\|(\lambda I + K_\alpha)^{-1}\|_{L_2} \leq C, \tag{3.3.35}$$

where C is independent of α. Now we have also shown, in Lemma 3.3.7 that $\|(\lambda I + K_\alpha)^{-1}\|_{L_2}$ depends continuously on α. Hence for all $\alpha \in [1, A]$ we have

$$\|(\lambda I + K_\alpha)^{-1}\|_{L_2} \leq C'. \tag{3.3.36}$$

Combining (3.3.35) and (3.3.36) we get

$$\|(\lambda I + K_\alpha)^{-1}\|_{L_2} \leq \max\{C, C'\},$$

and hence the result. □

We now just need one more result before we can prove the main result of this section - Theorem 3.3.10.

Lemma 3.3.9 *There exists a constant C independent of n and α such that*

$$\|(I - \mathcal{P}_n)K_\alpha\|_{L_2} \leq Cn^{-1}.$$

Proof: Let $v \in L_2$. Then by (3.3.27) and Lemma 3.2.7, we have

$$\|(I - \mathcal{P}_n)K_\alpha v\|_{L_2} \leq Cn^{-1}\|K_\alpha v\|_{H^1} \leq Cn^{-1}\|v\|_{L_2}$$

with C independent of α, and the result follows. □

We are now in a position to finally complete the proof of Theorem 3.3.1.

Theorem 3.3.10 *For n sufficiently large, and for all $\alpha \geq 1$, $(\lambda I + \mathcal{P}_n K_\alpha)^{-1}$ exists, and there exists a constant C independent of n and α such that*

$$\|(\lambda I + \mathcal{P}_n K_\alpha)^{-1}\|_{L_2} \leq C.$$

Proof: We use Theorem 3.3.2 with $S = (\lambda I + K_\alpha)$, and $T = (\lambda I + \mathcal{P}_n K_\alpha)$. Then

$$\|S - T\|_{L_2} = \|(I - \mathcal{P}_n)K_\alpha\|_{L_2} \leq Cn^{-1},$$

where C is independent of α, by Lemma 3.3.9. Now by Lemmas 3.3.5 and 3.3.8 we have that $(\lambda I + K_\alpha)^{-1}$ exists, and that $\|(\lambda I + K_\alpha)^{-1}\|_{L_2}$ is bounded independently of α. So, for n large enough that

$$\|(I - \mathcal{P}_n)K_\alpha\|_{L_2} \leq \frac{1}{\|(\lambda I + K_\alpha)^{-1}\|_{L_2}},$$

we have that S and T satisfy the assumptions of Theorem 3.3.2, and hence $(\lambda I + \mathcal{P}_n K_\alpha)^{-1}$ exists, and

$$\|(\lambda I + \mathcal{P}_n K_\alpha)^{-1}\|_{L_2} \leq \frac{\|(\lambda I + K_\alpha)^{-1}\|_{L_2}}{1 - \|(\lambda I + K_\alpha)^{-1}\|_{L_2} C n^{-1}} \leq C \tag{3.3.37}$$

where C is independent of α. This completes the proof. □

3.4 Evaluating the integrals

In this section we address the question of how to evaluate the matrix entries in (3.1.14). This requires computation of integrals of the form $K_\alpha v(t)$ (3.2.22), where $v(\tau) \in T_n$, with T_n given by (3.1.9), and t is one of the node points $t_j^{(n)}$. These integrals are hard to evaluate because as α grows the integrand becomes very highly peaked near $t = \tau$. In Figure 3-1 we plot the integrand in (3.2.22) for the case $v = 1$, $t = \pi$, and γ the unit circle, and for $\alpha = 10$ and $\alpha = 100$. One can clearly see how the integrand becomes more peaked as α increases, although

both graphs have the same peak value (approximately -0.08). It can be easily shown (see [1, (9.6.11)] and Appendix B) that $\Psi(0) = -2/\pi$, and when γ is the unit circle, $w(t,\tau) = 1/8$. Hence in this case the peak value of the integrand is $-1/4\pi \approx -0.08$ regardless of the value of α.

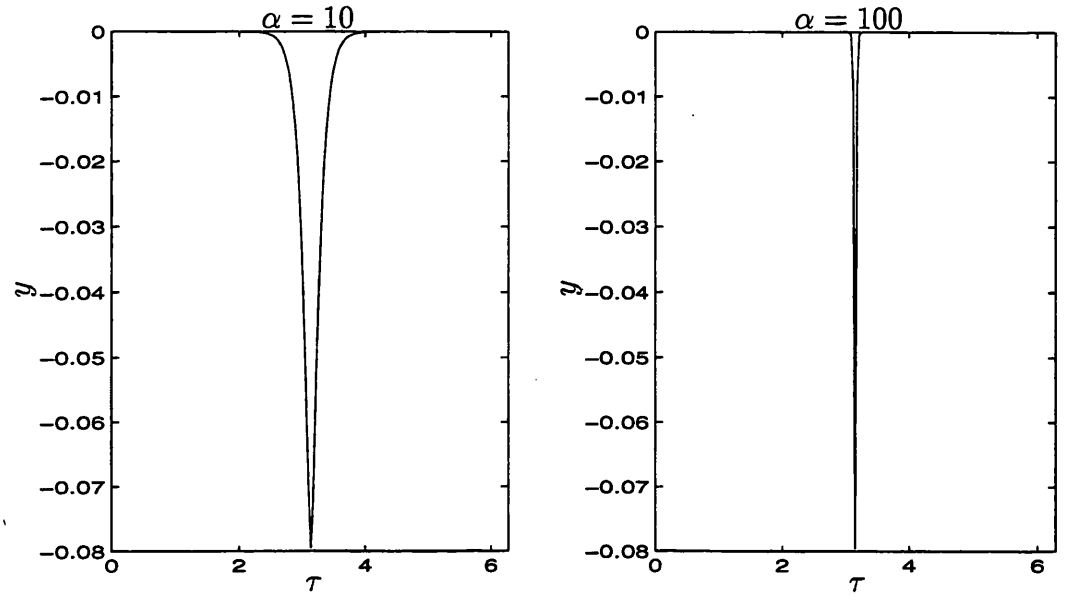

Figure 3-1: Graph of $y(\tau) := \Psi(\alpha r(\pi,\tau))w(\pi,\tau)$ against $\tau \in [0, 2\pi]$.

The basic idea of our method is to use a transformation to stretch the region of integration, and to remove the α-dependence of the kernel. We can then use a standard quadrature rule on a graded mesh to evaluate the integral.

To evaluate (3.2.22) we make the transformation

$$\tau = t - \frac{\tilde{\tau}}{\alpha}$$

to get

$$(K_\alpha v)(t) = \frac{1}{\alpha}\int_{-\alpha\pi}^{\alpha\pi} \Psi\left(\alpha r\left(t, t - \frac{\tilde{\tau}}{\alpha}\right)\right) w\left(t, t - \frac{\tilde{\tau}}{\alpha}\right) v\left(t - \frac{\tilde{\tau}}{\alpha}\right) d\tilde{\tau}. \quad (3.4.38)$$

We can then split (3.4.38) into the integral over $[0, \alpha\pi]$ plus the integral over $[-\alpha\pi, 0]$, and then, making the substitution $\tau = -\tilde{\tau}$ in the second integral, and

removing the tildes for clarity, we get:-

$$(K_\alpha v)(t) = \frac{1}{\alpha}\int_0^{\alpha\pi} \left\{\Psi\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right) w\left(t, t-\frac{\tau}{\alpha}\right) v\left(t-\frac{\tau}{\alpha}\right)\right. \quad (3.4.39)$$
$$\left. +\Psi\left(\alpha r\left(t, t+\frac{\tau}{\alpha}\right)\right) w\left(t, t+\frac{\tau}{\alpha}\right) v\left(t+\frac{\tau}{\alpha}\right)\right\} d\tau.$$

Observe the effect of this transformation on the integral (3.2.22). As α gets large, the region of integration gets stretched, but the derivatives of the integrand with respect to τ (away from $\tau = 0$) are now small as $\alpha \to \infty$. Now note that w and v are both smooth functions, and that $\Psi(x)$ decays exponentially as $x \to \infty$ and has a logarithmic singularity in its second order derivative at $x = 0$. From this it is clear that the integrand in (3.4.39) (as a function of τ) has a mild singularity near $\tau = 0$ and is exponentially decaying away from $\tau = 0$ when α is large. An effective method for evaluating this sort of integral is the graded mesh technique developed in [63], and used in [32].

First, we define the composite mesh $\Pi_m = x_0, x_1, \ldots, x_{m-1}$, on $[0, \infty)$, which consists of the points

$$\left[\frac{i}{N}\right]^{q_1}, \quad i = 0, \ldots, N, \quad (3.4.40)$$

for some $q_1 \geq 1$, $N \in \mathbb{N}$, together with the points

$$\left\{q_2 \log\left[\frac{rN}{rN-k}\right], k = 1, \ldots, rN-1\right\} \cap (1, \alpha\pi] \quad (3.4.41)$$

for some $q_2 > 0$ and $r \in \mathbb{N}$. If the last point is less than $\alpha\pi$, we add an extra point at $\alpha\pi$ to complete the mesh, thus ensuring that $x_{m-1} = \alpha\pi$. Note that the mesh parameters q_1, q_2, and r can be varied, to adjust the mesh. Increasing q_1 bunches the points on $[0, 1]$ closer to zero, increasing q_2 spreads out the points on $(1, \alpha\pi)$, and increasing r increases the ratio of the number of points in $(1, \alpha\pi)$ to the number of points in $[0, 1]$.

Now for any $f : [0, \infty) \mapsto \mathbb{R}$, let $\mathcal{Q}_m f$ denote the function which is linear on each $[x_{i-1}, x_i]$, and which interpolates f at each x_i, $i = 0, \ldots, m-1$, with

$\mathcal{Q}_m f \equiv 0$ on (x_{m-1}, ∞). Then define a quadrature approximation to $(K_\alpha v)(t)$ by

$$\begin{aligned}(K_{\alpha,m}v)(t) &= \frac{1}{\alpha}\int_0^{\alpha\pi} \mathcal{Q}_m\left\{\Psi\left(\alpha r\left(t, t-\frac{(\cdot)}{\alpha}\right)\right) w\left(t, t-\frac{(\cdot)}{\alpha}\right)\right\}(\tau) v\left(t-\frac{\tau}{\alpha}\right) \\ &\quad + \mathcal{Q}_m\left\{\Psi\left(\alpha r\left(t, t+\frac{(\cdot)}{\alpha}\right)\right) w\left(t, t+\frac{(\cdot)}{\alpha}\right)\right\}(\tau) v\left(t+\frac{\tau}{\alpha}\right) d\tau. \end{aligned} \tag{3.4.42}$$

For the computation of the matrix in (3.1.14) we shall replace K_α by $K_{\alpha,m}$ and thus we require in practice the evaluation of $K_{\alpha,m}\phi_k$, with ϕ_k given in (3.1.10). This is easily realised in practice since it requires the evaluation only of integrals of the form

$$\int_{x_{j-1}}^{x_{j+1}} \psi_j(\tau)\phi_k\left(t \pm \frac{\tau}{\alpha}\right) d\tau,$$

where ψ_j, $j = 1, \ldots, m-2$ are the "hat" functions on $[x_{j-1}, x_{j+1}]$ centred at x_j. These integrals are easy to evaluate analytically.

This completes the description of the method employed to evaluate the integrals needed in the linear system (3.1.14), we shall now present an error analysis, leading to the main result of this section, which is given in Theorem 3.4.4 below.

Before getting this result we first need to make a few definitions:- Let $l \in \mathbb{N}$, $\mu > 0$ and $0 < \beta \leq l$. Then for any $v : \mathbb{R}^+ \mapsto \mathbb{C}$ define

$$\|v\|_{l,\beta} = \max\left\{\sup_{x\in[0,1]} |v(x)|, \sup_{x\in[0,1]} |x^{[j-\beta]}D^j v(x)|,\ j = 1, \ldots, l\right\},$$

where

$$[j-\beta] = \begin{cases} j-\beta, & \text{if } j \geq \beta \\ 0, & \text{if } j \leq \beta \end{cases}$$

and for any $v : \mathbb{R}^+ \mapsto \mathbb{C}$ define

$$\|v : e^{-\mu x}\|_l = \max\left\{\sup_{x\in[1,\infty)} |e^{\mu x}D^j v(x)|,\ j = 0, \ldots, l\right\}.$$

Now set

$$\|v : e^{-\mu x}\|_{l,\beta} = \max\left\{\|v\|_{l,\beta},\ \|v : e^{-\mu x}\|_l\right\}. \tag{3.4.43}$$

Then we make the following definition:-

Definition 3.4.1 *Let $C^l_\beta[e^{-\mu x}]$ denote the set of all functions $v : \mathbb{R}^+ \mapsto \mathbb{C}$ which have l continuous derivatives on $(0,\infty)$ and for which $\|v : e^{-\mu x}\|_{l,\beta} < \infty$.*

Note that functions in the class $C^l_\beta[e^{-\mu x}]$ have l derivatives which "behave like" x^β as $x \to 0$ and "behave like" $e^{-\mu x}$ as $x \to \infty$.

We can now quote the following theorem, which appears in a more general form as Theorem 2.6 in [63]. It concerns the approximation power of the interpolation operator $\mathcal{Q}_m$ on the mesh Π_m given by (3.4.40) and (3.4.41).

Theorem 3.4.2 *There exists a constant C independent of α and m such that for any $v \in C^2_\beta[e^{-\mu x}]$, $0 < \beta < 1$, $\mu > 0$,*

$$\|(I - \mathcal{Q}_m)v\|_{L_\infty[0,\alpha\pi]} \leq C\frac{1}{m^2}\|v : e^{-\mu x}\|_{2,\beta},$$

provided $q_1 \geq 2/\beta$ and $q_2 \geq 2/\mu$.

Proof: See [63, p.30].

Now, in our quadrature approximation $K_{\alpha,m}$, given by (3.4.42), we are applying the operator $\mathcal{Q}_m$ to the functions

$$f_{\alpha,t,-}(\tau) = \Psi\left(\alpha r\left(t, t - \tau/\alpha\right)\right) w\left(t, t - \tau/\alpha\right)$$

and

$$f_{\alpha,t,+}(\tau) = \Psi\left(\alpha r\left(t, t + \tau/\alpha\right)\right) w\left(t, t + \tau/\alpha\right).$$

We shall use Theorem 3.4.2 to analyse the error in this approximation. In order to do this we need to establish that these functions fit the assumptions of Theorem 3.4.2. Since the analysis is exactly the same for $f_{\alpha,t,-}$ and $f_{\alpha,t,+}$, for simplicity we will just present it for the former.

Theorem 3.4.3 *There exists a constant C independent of t and $\alpha \geq 1$ such that*

$$\|f_{\alpha,t,-} : e^{-\mu\tau}\|_{2,\beta} \leq C$$

for all $0 < \beta \leq 1$, $0 < \mu < C_1$, where C_1 is the constant on the left hand side of (3.2.25) (note that by Proposition 3.2.4, $C_1 > 0$).

Proof: First, note the following:

(i) By Propositions 3.2.2(ii) and 3.2.3(ii), Ψ and Ψ' are both bounded on $[0, \infty)$. As a result, there exist constants C_3 and C_4 independent of α and $t, \tau \in \mathbb{R}$ such that

$$\left|\Psi\left(\alpha r\left(t, t - \frac{\tau}{\alpha}\right)\right)\right| \leq C_3$$

and

$$\left|\Psi'\left(\alpha r\left(t, t - \frac{\tau}{\alpha}\right)\right)\right| \leq C_4.$$

(ii) γ and all its derivatives are C^∞ and 2π-periodic, and thus $|\gamma(t)|$ is continuous and 2π-periodic in t. Hence, $\sup_{t\in[0,2\pi]} |\gamma(t)| \leq C$, and so $\sup_{t\in\mathbb{R}} |\gamma(t)| \leq C$. Thus, for all $t, \tau \in \mathbb{R}$,

$$|r(t,\tau)| = r(t,\tau) = |\gamma(t) - \gamma(\tau)| \leq |\gamma(t)| + |\gamma(\tau)| \leq C.$$

(iii) $w(t,\tau)$ is also 2π-periodic in t, τ, and (see (3.2.23)) infinitely continuously differentiable, hence

$$\sup_{t,\tau\in\mathbb{R}} \left|\frac{\partial^j}{\partial\tau^j}\left\{w\left(t, t - \frac{\tau}{\alpha}\right)\right\}\right| \leq C$$

independently of $\alpha \geq 1$, for all $j = 0, 1, 2, \ldots$.

(iv) Noting that $r(t,\tau) = r(\tau,t)$, we have from Proposition 3.2.1 that there exists a constant C such that

$$\left|\frac{\partial}{\partial\tau} r(t,\tau)\right| \leq C, \quad \text{for all } t, \tau \in [0, 2\pi].$$

Throughout the rest of this proof we will freely use facts (i)-(iv) above without

explicitly referring to them. Now, recalling the definition (3.4.43),

$$\|f_{\alpha,t,-} : e^{-\mu\tau}\|_{2,\beta} = \max\{\|f_{\alpha,t,-}\|_{2,\beta}, \|f_{\alpha,t,-} : e^{-\mu\tau}\|_2\}.$$

First we will show that $\|f_{\alpha,t,-}\|_{2,\beta}$ is bounded independently of α and t for all $\beta \leq 1$, by looking at each of the terms on the right-hand side of the following expression:

$$\|f_{\alpha,t,-}\|_{2,\beta} = \max\left\{ \underbrace{\sup_{\tau\in[0,1]} |f_{\alpha,t,-}(\tau)|}_{(a)}, \underbrace{\sup_{\tau\in[0,1]} |\tau^{1-\beta}\frac{\partial}{\partial\tau} f_{\alpha,t,-}(\tau)|}_{(b)}, \underbrace{\sup_{\tau\in[0,1]} |\tau^{2-\beta}\frac{\partial^2}{\partial\tau^2} f_{\alpha,t,-}(\tau)|}_{(c)} \right\}.$$

We will bound each of (a), (b), (c) in turn. First (a) and (b), which are relatively easy.

(a) For all α and $t \in [0, \alpha\pi]$

$$\begin{aligned} \sup_{\tau\in[0,1]} |f_{\alpha,t,-}(\tau)| &\leq \sup_{\tau\in[0,1]} \left|\Psi\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right| \sup_{\tau\in[0,1]} \left|w\left(t, t-\frac{\tau}{\alpha}\right)\right| \\ &\leq C. \end{aligned}$$

(b) Using the chain rule and proceeding analogously

$$\begin{aligned} &\sup_{\tau\in[0,1]} \left|\tau^{1-\beta}\frac{\partial}{\partial\tau} f_{\alpha,t,-}(\tau)\right| \\ &\leq \sup_{\tau\in[0,1]} \left\{ \left|\Psi\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right| \left|\frac{\partial}{\partial\tau}\left\{w\left(t, t-\frac{\tau}{\alpha}\right)\right\}\right| \right. \\ &\qquad \left. + \alpha \left|w\left(t, t-\frac{\tau}{\alpha}\right)\right| \left|\frac{\partial}{\partial\tau}\left\{r\left(t, t-\frac{\tau}{\alpha}\right)\right\}\right| \left|\Psi'\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right| \right\} \qquad (3.4.44) \\ &\leq C, \end{aligned}$$

where the final inequality follows since (with $\tilde{\tau} = t - \tau/\alpha$),

$$\left|\frac{\partial}{\partial\tau}\left\{r\left(t, t-\frac{\tau}{\alpha}\right)\right\}\right| = \left|-\frac{1}{\alpha}\frac{\partial}{\partial\tilde{\tau}} r(t,\tilde{\tau})\right| \leq \frac{C}{\alpha}. \qquad (3.4.45)$$

(c) Similarly, we have

$$\begin{aligned}\left|\frac{\partial^2}{\partial\tau^2}f_{\alpha,t,-}(\tau)\right| &\leq \left|\Psi\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\right|\left|\frac{\partial^2}{\partial\tau^2}\left\{w\left(t,t-\frac{\tau}{\alpha}\right)\right\}\right| \\ &\quad + 2\left|\frac{\partial}{\partial\tau}\left\{\Psi\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\right\}\right|\left|\frac{\partial}{\partial\tau}\left\{w\left(t,t-\frac{\tau}{\alpha}\right)\right\}\right| \\ &\quad + \left|\frac{\partial^2}{\partial\tau^2}\left\{\Psi\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\right\}\right|\left|w\left(t,t-\frac{\tau}{\alpha}\right)\right| \\ &\leq C\left[1+\left|\frac{\partial^2}{\partial\tau^2}\left\{\Psi\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\right\}\right|\right]. \qquad (3.4.46)\end{aligned}$$

Now, to bound the last term we must proceed more carefully. First note that

$$\frac{\partial}{\partial\tau}\left\{\Psi\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\right\} = \Psi'\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\frac{\partial}{\partial\tau}\left\{\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right\}. \quad (3.4.47)$$

It is clear from (3.2.24) that $\frac{\partial}{\partial\tau}r(t,\tau)$ is discontinuous as $t\to\tau$. For example on the unit circle $\gamma(t)=(\cos t,\sin t)^T$, $\gamma(\tau)=(\cos\tau,\sin\tau)^T$, we have (after a little calculation)

$$\frac{\partial}{\partial\tau}r(t,\tau)=\frac{\sin(t-\tau)}{2|\sin(t-\tau)/2|}\to\begin{cases}1 & \text{as } t\to\tau_+\\ -1 & \text{as } t\to\tau_-.\end{cases}$$

So, we can't differentiate (3.4.47) again at $\tau=0$. However, for all $\tau>0$ the derivative is well defined, and, in that case,

$$\begin{aligned}\frac{\partial^2}{\partial\tau^2}\left\{\Psi\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\right\} &= \Psi''\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\left[\frac{\partial}{\partial\tau}\left\{\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right\}\right]^2 \\ &\quad + \Psi'\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\frac{\partial^2}{\partial\tau^2}\left\{\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right\}.\end{aligned}$$

Using (3.4.45), we have

$$\begin{aligned}&\left|\tau^{2-\beta}\frac{\partial^2}{\partial\tau^2}\left\{\Psi\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\right\}\right| \\ &\leq C\left[\left|\tau^{2-\beta}\Psi''\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)\right|+\left|\tau^{2-\beta}\frac{\partial^2}{\partial\tau^2}\left\{\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right\}\right|\right].\end{aligned} \qquad (3.4.48)$$

Consider first $\Psi''(\alpha r(t, t-\frac{\tau}{\alpha}))$. Recalling the definition of $\Psi(x)$, it can be easily shown (see for example [1]) that $\Psi''(x) = \Psi(x) + iH_0^{(1)}(ix)$. Now since Ψ is bounded on $[0,\infty)$ we have

$$|\Psi''(x)| \leq C\{1 + |iH_0^{(1)}(ix)|\},$$

where (see [1]), $iH_0^{(1)}(ix)$ is real, positive valued, and decreasing on $[0,\infty)$ and blows up logarithmically as $x \to 0$. Moreover, from Proposition 3.2.4 we also know that there exists a constant C_1 independent of α such that

$$\alpha r\left(t, t-\frac{\tau}{\alpha}\right) \geq C_1\tau.$$

These last two expressions tell us that, for $\tau \in [0,1]$, $0 < \beta \leq 1$,

$$\begin{aligned}\left|\tau^{2-\beta}\Psi''\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right| &\leq C\left\{1 + \left|\tau^{2-\beta} iH_0^{(1)}\left(i\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right|\right\} \\ &\leq C\left\{1 + \tau^{2-\beta} iH_0^{(1)}(iC_1\tau)\right\} \leq C, \qquad (3.4.49)\end{aligned}$$

where C is independent of α.

This bounds the first term on the right hand side of (3.4.48). Consider now the second term. For $\tau \in (0,1]$,

$$\frac{\partial^2}{\partial\tau^2}\left\{\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right\} = \frac{1}{\alpha}\frac{\partial^2}{\partial\tilde{\tau}^2} r(t,\tilde{\tau}), \qquad (3.4.50)$$

with $\tilde{\tau}$ as above. From the definition

$$\frac{\partial}{\partial\tilde{\tau}} r(t,\tilde{\tau}) = \frac{(\gamma(\tilde{\tau}) - \gamma(t)).\gamma'(\tilde{\tau})}{|\gamma(t) - \gamma(\tilde{\tau})|} = \frac{(\gamma_1(\tilde{\tau}) - \gamma_1(t))\gamma_1'(\tilde{\tau}) + (\gamma_2(\tilde{\tau}) - \gamma_2(t))\gamma_2'(\tilde{\tau})}{|\gamma(t) - \gamma(\tilde{\tau})|}.$$

It follows that for $\tilde{\tau} < t$,

$$\frac{\partial^2}{\partial\tilde{\tau}^2} r(t,\tilde{\tau}) = -\frac{\{(\gamma(\tilde{\tau}) - \gamma(t)).\gamma'(\tilde{\tau})\}^2}{|\gamma(t) - \gamma(\tilde{\tau})|^3} + \frac{|\gamma'(\tilde{\tau})|^2 + (\gamma(\tilde{\tau}) - \gamma(t)).\gamma''(\tilde{\tau})}{|\gamma(t) - \gamma(\tilde{\tau})|}.$$

Hence

$$\left|\frac{\partial^2}{\partial\tilde{\tau}^2} r(t,\tilde{\tau})\right| \leq C|\gamma(t) - \gamma(\tilde{\tau})|^{-1} \leq C|t - \tilde{\tau}|^{-1},$$

(recalling Proposition 3.2.4) and so, for $\tau \in (0,1]$, (3.4.50) yields

$$\frac{\partial^2}{\partial \tau^2}\left\{\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right\} \leq \frac{1}{\alpha} C|t-\tilde{\tau}|^{-1} \leq C\frac{1}{\tau}.$$

This gives us, for $\tau \in [0,1]$, $0 < \beta \leq 1$,

$$\tau^{2-\beta}\frac{\partial^2}{\partial \tau^2}\left\{\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right\} \leq C\tau^{1-\beta} \leq C. \tag{3.4.51}$$

Inserting (3.4.49) and (3.4.51) into (3.4.48) and recalling (3.4.46) gives us the result that $\|f_{\alpha,t,-}\|_{2,\beta}$ is bounded independently of t and α.

Now it just remains to show that $\|f_{\alpha,t,-} : e^{-\mu\tau}\|_2$ is bounded independently of α and t for all $\mu \in (0, C_1)$. This turns out to be quite straightforward. First write

$$\|f_{\alpha,t,-} : e^{-\mu\tau}\|_2 =$$

$$\max\left\{\underbrace{\sup_{\tau\in[1,\infty)}|e^{\mu\tau}f_{\alpha,t,-}(\tau)|}_{(d)}, \underbrace{\sup_{\tau\in[1,\infty)}|e^{\mu\tau}\frac{\partial}{\partial\tau}f_{\alpha,t,-}(\tau)|}_{(e)}, \underbrace{\sup_{\tau\in[1,\infty)}|e^{\mu\tau}\frac{\partial^2}{\partial\tau^2}f_{\alpha,t,-}(\tau)|}_{(f)}\right\}.$$

We bound each of these terms in turn.

(d) Using Proposition 3.2.2(ii) and Proposition 3.2.4 we have for all $\rho \in [0,1)$,

$$\left|\Psi\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right| \leq Ce^{-\rho\left(\alpha r\left(t,t-\frac{\tau}{\alpha}\right)\right)} \leq Ce^{-\rho C_1\tau}.$$

Hence if $\mu \in (0, C_1)$ then for all $\tau \in [1,\infty)$,

$$|e^{\mu\tau}f_{\alpha,t,-}(\tau)| \leq Ce^{\tau(\mu-\rho C_1)}.$$

By choosing $\rho = \mu/C_1 \in (0,1)$ we obtain the result that (d) is bounded independently of α and t.

(e) Using (3.4.44) we have, for all $\tau \in [1,\infty)$,

$$\left|\frac{\partial}{\partial\tau}f_{\alpha,t,-}(\tau)\right| \leq C\left\{\left|\Psi\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right| + \left|\Psi'\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right|\right\}.$$

Now using Propositions 3.2.2(ii) and 3.2.3(ii), the same argument as was used in (d) can be applied to show that (e) is bounded independently of α and t.

(f) From the first inequality on the right hand side of (3.4.46), we get for $\tau \in [1, \infty)$

$$\begin{aligned}&\left|\frac{\partial^2}{\partial \tau^2} f_{\alpha,t,-}(\tau)\right| \\ &\leq C\left\{\left|\Psi\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right| + \left|\Psi'\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right| + \left|\Psi''\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right)\right|\right\}.\end{aligned}$$

The estimation of this follows the same lines as that in (d), (e), since from Appendix B and [1, (9.7.2)] it can be deduced that

$$|\Psi''(x)| \leq Be^{-\rho x},$$

for all $\rho \in [0, 1)$. This completes the proof of Theorem 3.4.3. □

We are now in a position to prove the main result of this section.

Theorem 3.4.4 *For all $v \in L_2$ there exists a constant C independent of α and m such that*

$$\|K_{\alpha,m} v - K_\alpha v\|_{L_\infty} \leq C \frac{1}{m^2} \|v\|_{L_2}.$$

Proof: By (3.4.39), (3.4.42)

$$
\begin{aligned}
&\|K_{\alpha,m}v - K_\alpha v\|_{L_\infty} \\
&= \sup_{t\in[0,2\pi]} \left| \frac{1}{\alpha}\int_0^{\alpha\pi} \left\{ (\mathcal{Q}_m - I)\{f_{\alpha,t,-}(\cdot)\}(\tau) v\left(t - \frac{\tau}{\alpha}\right) \right.\right. \\
&\qquad \left.\left. + (\mathcal{Q}_m - I)\{f_{\alpha,t,+}(\cdot)\}(\tau) v\left(t + \frac{\tau}{\alpha}\right) \right\} d\tau \right| \\
&\leq \sup_{t\in[0,2\pi]} \frac{1}{\alpha} \left\{ \int_0^{\alpha\pi} |(\mathcal{Q}_m - I)\{f_{\alpha,t,-}(\cdot)\}(\tau)|^2 d\tau \right\}^{\frac{1}{2}} \left\{ \int_0^{\alpha\pi} \left| v\left(t - \frac{\tau}{\alpha}\right)\right|^2 d\tau \right\}^{\frac{1}{2}} \\
&\quad + \sup_{t\in[0,2\pi]} \frac{1}{\alpha} \left\{ \int_0^{\alpha\pi} |(\mathcal{Q}_m - I)\{f_{\alpha,t,+}(\cdot)\}(\tau)|^2 d\tau \right\}^{\frac{1}{2}} \left\{ \int_0^{\alpha\pi} \left| v\left(t + \frac{\tau}{\alpha}\right)\right|^2 d\tau \right\}^{\frac{1}{2}} \\
&\leq \sup_{t\in[0,2\pi]} \frac{1}{\alpha} \left\{ \alpha\pi \| [(\mathcal{Q}_m - I)\{f_{\alpha,t,-}(\cdot)\}(\tau)]^2 \|_{L_\infty[0,\alpha\pi]} \right\}^{\frac{1}{2}} \left\{ \alpha \int_{t-\pi}^{t} |v(\tau)|^2 d\tau \right\}^{\frac{1}{2}} \\
&\quad + \sup_{t\in[0,2\pi]} \frac{1}{\alpha} \left\{ \alpha\pi \| [(\mathcal{Q}_m - I)\{f_{\alpha,t,+}(\cdot)\}(\tau)]^2 \|_{L_\infty[0,\alpha\pi]} \right\}^{\frac{1}{2}} \left\{ \alpha \int_{t}^{t+\pi} |v(\tau)|^2 d\tau \right\}^{\frac{1}{2}} \\
&\leq C\frac{1}{m^2}\|v\|_{L_2},
\end{aligned}
$$

where the last line follows by Theorems 3.4.2 and 3.4.3. This completes the proof. □

3.5 Fully discrete collocation

With $K_{\alpha,m}$ defined as in §3.4, the fully discrete collocation method for solving (3.1.6) is then to find $u_{n,m} \in T_n$ satisfying

$$(\lambda I + K_{\alpha,m})u_{n,m}(t_j^{(n)}) = f(t_j^{(n)}), \quad j = 0, \ldots, 2n-1. \tag{3.5.52}$$

In this section we will present a fully discrete error analysis for this method. We start by quoting the following theorem, which appears in a much more general form as Theorem 1 in [79].

Theorem 3.5.1 *There exists a constant C independent of n such that for all $f \in C[0, 2\pi]$ the interpolatory projection $\mathcal{P}_n : C \to T_n$ given by (3.1.12) satisfies*

$$\|\mathcal{P}_n f\|_{L_2} \leq C\|f\|_{L_\infty}.$$

Proof: For a full proof of the more general theorem see [79]. This more general theorem in fact proves the result for certain types of discrete orthogonal projection. The fact that the interpolatory projection $\mathcal{P}_n$ is also a discrete orthogonal projection is well known, see for example [6, Lemma 2]. □

We can now use this theorem to prove the following result.

Lemma 3.5.2 *For n and m sufficiently large there exists a constant C independent of α, m and n such that*

$$\|\mathcal{P}_n(K_{\alpha,m} - K_\alpha)v\|_{L_2} \leq C\frac{1}{m^2}\|v\|_{L_2}, \quad \text{for all } v \in L_2.$$

Proof: By Theorems 3.5.1 and 3.4.4,

$$\|\mathcal{P}_n(K_{\alpha,m} - K_\alpha)v\|_{L_2} \leq C\|(K_{\alpha,m} - K_\alpha)v\|_{L_\infty} \leq C\frac{1}{m^2}\|v\|_{L_2}. \qquad \square$$

We now need one more result before we can establish stability.

Lemma 3.5.3 *For n and m sufficiently large there exists a constant C independent of α, m and n such that*

$$\|\mathcal{P}_n K_{\alpha,m} - K_\alpha\|_{L_2} \leq C\left[\frac{1}{m^2} + \frac{1}{n}\right].$$

Proof:

$$\begin{aligned}
\|\mathcal{P}_n K_{\alpha,m} - K_\alpha\|_{L_2} &= \|\mathcal{P}_n K_{\alpha,m} - \mathcal{P}_n K_\alpha + \mathcal{P}_n K_\alpha - K_\alpha\|_{L_2} \\
&\leq \|\mathcal{P}_n(K_{\alpha,m} - K_\alpha)\|_{L_2} + \|\mathcal{P}_n K_\alpha - K_\alpha\|_{L_2} \\
&\leq C\left\{\frac{1}{m^2} + \frac{1}{n}\right\}
\end{aligned}$$

where the last line follows by Lemma 3.5.2 and Lemma 3.3.9. □

The norm convergence established in Lemma 3.5.3 can now be used to prove that the method is stable.

Theorem 3.5.4 *For n and m sufficiently large, and for $\alpha \geq 1$, $(\lambda I + \mathcal{P}_n K_{\alpha,m})^{-1}$ exists, and is bounded in $\|\cdot\|_{L_2}$ independently of α, n and m.*

Proof: Since

$$\|(\lambda I + K_\alpha) - (\lambda I + \mathcal{P}_n K_{\alpha,m})\|_{L_2} = \|\mathcal{P}_n K_{\alpha,m} - K_\alpha\|_{L_2} \to 0 \quad \text{as } n, m \to \infty,$$

uniformly in α by Lemma 3.5.3, the result follows by using Theorem 3.3.2 with $S = (\lambda I + K_\alpha)$, and $T = (\lambda I + \mathcal{P}_n K_{\alpha,m})$, and Lemmas 3.3.5 and 3.3.8 as in the proof of Theorem 3.3.10. □

Finally, we prove the convergence of the method.

Theorem 3.5.5 *For n and m sufficiently large, and for $\alpha \geq 1$, the discrete collocation equation (3.5.52) has a unique solution $u_{n,m} \in T_n$, and we have the following error estimate, where u is the solution of (3.1.6), and the constant C is independent of α, n and m:-*

$$\|u - u_{n,m}\|_{L_2} \leq C \left[\frac{1}{n^p} \|u\|_{H^p} + \frac{1}{m^2} \|u\|_{L_2} \right], \tag{3.5.53}$$

provided q_1, q_2 are chosen to satisfy $q_1 \geq 2$ and $q_2 > 2/C_1$, with C_1 as in (3.2.25).

Proof: Noting that two functions coincide at the $t_j^{(n)}$ if and only if their interpolants coincide, and, since $\mathcal{P}_n u_{n,m} = u_{n,m}$, (3.5.52) is equivalent to

$$(\lambda I + \mathcal{P}_n K_{\alpha,m}) u_{n,m} = \mathcal{P}_n f.$$

By Theorem 3.5.4 we then have

$$u_{n,m} = (\lambda I + \mathcal{P}_n K_{\alpha,m})^{-1} \mathcal{P}_n f.$$

Then a little algebra shows

$$u - u_{n,m} = (\lambda I + \mathcal{P}_n K_{\alpha,m})^{-1} [\lambda (u - \mathcal{P}_n u) - \mathcal{P}_n (K_\alpha - K_{\alpha,m}) u],$$

and by Theorem 3.5.4

$$\begin{aligned} \|u - u_{n,m}\|_{L_2} &\leq C \|(\lambda I + \mathcal{P}_n K_{\alpha,m})^{-1}\|_{L_2} [\|u - \mathcal{P}_n u\|_{L_2} + \|\mathcal{P}_n (K_\alpha - K_{\alpha,m}) u\|_{L_2}] \\ &\leq C \left[\frac{1}{n^p} \|u\|_{H^p} + \frac{1}{m^2} \|u\|_{L_2} \right] \end{aligned}$$

where the last line follows by (3.3.27) and Lemma 3.5.2. This proves the convergence rate of our method. □

3.6 Numerical examples

In this section, as an example, we solve the second kind integral equation

$$-\frac{1}{2}u + K_\alpha u = 1 \tag{3.6.54}$$

using the method (3.5.52). This integral equation arises from solving (3.1.1) on a bounded domain $\Omega \subset \mathbb{R}^2$ with Dirichlet data $U = 1$ on the boundary, and then the solution U of (3.1.1) can be expressed as

$$U = \mathcal{K}_\alpha u \tag{3.6.55}$$

where u is the solution of (3.6.54).

As domain boundaries we choose the simple ellipse

$$\gamma(t) = (5\cos t, \sin t), \quad t \in [0, 2\pi], \tag{3.6.56}$$

and then the more complicated boundary curve

$$\gamma(t) = (\cos t(1 + \cos^2 4t), \sin t(1 + \cos^2 4t)), \quad t \in [0, 2\pi], \tag{3.6.57}$$

which is illustrated in Figure 3-2.

It is the ability of the boundary integral method to handle complicated regions such as this which makes it attractive for this problem. Note that when computing the matrix entries arising from method (3.5.52) for boundary curve (3.6.57), a Taylor series expansion must be used to evaluate the function $w(t, \tau)$ near $t = \tau$, whereas for (3.6.56) we can explicitly separate out the zeros in the numerator and the denominator, and thus compute $w(t, \tau)$ analytically.

For each domain we solve (3.6.54) for $\alpha = 10$ and $\alpha = 100$. For each experiment the quadrature points are as given in (3.4.40) and (3.4.41) with $q_1 = 2$, $q_2 = \alpha$ and $r = 10$. This choice of q_1 is obvious from Theorem 3.5.5, but the choices of q_2 and r are less clear. From Theorem 3.5.5 it is apparent that we

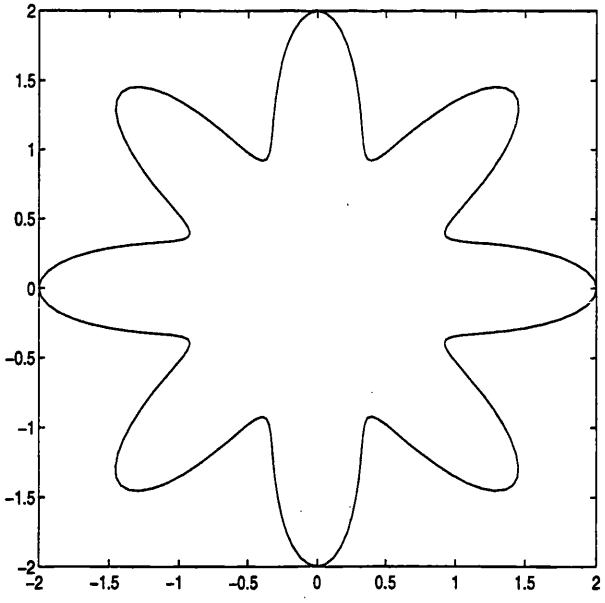

Figure 3-2: Boundary curve (3.6.57).

should choose $q_2 > 2/C_1$, with C_1 as in (3.2.25), but in practice the value of C_1 is unknown. Hence the values of q_2 and r were determined experimentally, taking into account the fact that as α increases the domain of integration $[0, \alpha\pi]$ gets stretched, but the derivatives of the integrand with respect to τ become smaller away from $\tau = 0$, and hence we can afford to have a greater mesh spacing on $[1, \alpha\pi]$. As q_2 depends on α, our results do not fully support the theory of §3.5 (as the constant C on the right hand side of (3.5.53) depends on q_2 and hence on α). However, we emphasise that this choice of q_2 was merely determined experimentally, and choosing a fixed value of q_2 independently of α should still lead to good results.

In each case we evaluate the approximate value of the density function $u_{n,m}$ at $t = 0$ and $t = \pi/3$. The exact solution is not known, so we compute a "near exact" solution by applying the method with a large (specified below) number of collocation and quadrature points, and then for the purpose of computing errors we treat this as the exact solution. For each example we double the number of quadrature points N on the interval $[0, 1]$, leading to an approximate doubling of the total number of quadrature points m on the interval $[0, \alpha\pi]$, and at the same time we increase the number of collocation points n logarithmically in N,

ie. $n = C \log N$. The estimated order of convergence (EOC) is then computed by

$$EOC = \log_2 \left(\frac{\text{error}_{2N}}{\text{error}_N} \right). \tag{3.6.58}$$

Now recalling (3.5.53), we have proved that in $\|\cdot\|_{L_2}$ the convergence is superalgebraic in n and second order in N. However, if γ is analytic we would expect (see e.g. [49, Chapters 11–12]) exponential convergence in $\|\cdot\|_{L_2}$ with respect to n, and we would hope that the pointwise error would be bounded in a similar way. Under these assumptions, we would expect that (provided $q_2 > 2/C_1$) EOC should be approximately equal to two. Results are given in Tables 3.2–3.5.

In Table 3.2 the "exact" solution is computed taking $n = 128$, $N = 640$, and $m = 6156$. The expected order of convergence (3.5.53) is clearly achieved.

In Table 3.3 the "exact" solution is computed taking $n = 128$, $N = 640$, and $m = 6702$. Again, the expected order of convergence (3.5.53) is achieved, demonstrating the robustness of the method with respect to α. These results should be compared with the poor results displayed in Table 3.1 for the method which is not robust to large α.

In Tables 3.4 and 3.5 the "exact" solution is computed taking $n = 224$, $N = 128$, and $m = 1233$ and $m = 1342$ respectively. Many more basis functions are needed for this example, due to the larger curvature of the boundary curve. Although the convergence is more erratic than for boundary curve (3.6.56), the estimated order of convergence is still broadly achieved.

On the more complicated boundary (3.6.57) we also evaluate (3.6.55) using the trapezoidal rule with $2n$ nodes, to compute the solution of the Helmholtz problem (3.1.1) with boundary data $U = 1$ at the points $(0, 0)$ and $(1, 1)$. Again we compute errors by applying the method with a large number of collocation and quadrature points, and take this to be the "exact" solution. Results are given in Tables 3.6 and 3.7. For $\alpha = 100$ the solution of (3.1.1) will be zero almost everywhere, and highly peaked in a boundary layer. The computed solution at the two points $(0, 0)$ and $(1, 1)$ can be seen to be nearly zero (Table 3.7), and so to confirm the convergence rate of our method we also compute the solution nearer the boundary, at the point $(1.95, 0)$. To compute the solution at this point we use the trapezoidal rule with $4n$ nodes. We need more nodes because the

α	n	N	m	t	$u_{n,m}(t)$	error(t)	EOC
10	10	10	98	0	-1.65391455	1.4e-3	-1.3
	12	20	194		-1.65316703	6.0e-4	-2.1
	15	40	386		-1.65271099	1.4e-4	-2.7
	18	80	771		-1.65258941	2.2e-5	-2.8
	21	160	1540		-1.65257093	3.2e-6	-3.4
	24	320	3079		-1.65256799	3.0e-7	
	128	640	6156		-1.65256769		
	10	10	98	$\pi/3$	-2.00230066	8.3e-3	-9.3
	12	20	194		-1.99395046	1.3e-5	-2.3
	15	40	386		-1.99396070	2.6e-6	-1.7
	18	80	771		-1.99396250	8.4e-7	-2.5
	21	160	1540		-1.99396319	1.5e-7	-1.9
	24	320	3079		-1.99396330	4.0e-8	
	128	640	6156		-1.99396334		

Table 3.2: Density function for the boundary curve (3.6.56), $\alpha = 10$ ($q_1 = 2$, $q_2 = 10$, $r = 10$ in (3.4.40)–(3.4.41)).

α	n	N	m	t	$u_{n,m}(t)$	error(t)	EOC
100	10	10	107	0	-1.95030178	1.1e-3	-2.0
	12	20	212		-1.95114131	2.7e-4	-2.1
	15	40	421		-1.95134900	6.6e-5	-2.2
	18	80	840		-1.95139990	1.5e-5	-2.5
	21	160	1677		-1.95141208	2.5e-6	-4.4
	24	320	3352		-1.95141471	1.2e-7	
	128	640	6702		-1.95141460		
	10	10	107	$\pi/3$	-2.00106129	1.7e-3	-8.2
	12	20	212		-1.99939058	5.9e-6	-2.0
	15	40	421		-1.99939492	1.5e-6	-2.0
	18	80	840		-1.99939606	3.8e-7	-2.1
	21	160	1677		-1.99939634	9.0e-8	-2.3
	24	320	3352		-1.99939642	1.8e-8	
	128	640	6702		-1.99939643		

Table 3.3: Density function for the boundary curve (3.6.56), $\alpha = 100$ ($q_1 = 2$, $q_2 = 100$, $r = 10$ in (3.4.40)–(3.4.41)).

α	n	N	m	t	$u_{n,m}(t)$	error(t)	EOC
10	64	4	41	0	-1.50636903	9.7e-3	-2.0
	96	8	79		-1.51358770	2.5e-3	-2.0
	128	16	156		-1.51542259	6.2e-4	-2.1
	160	32	310		-1.51589092	1.5e-4	-2.3
	192	64	618		-1.51600854	3.0e-5	
	224	128	1233		-1.51603822		
	64	4	41	$\pi/3$	-2.13906281	9.0e-3	-6.3
	96	8	79		-2.12994074	1.2e-4	-4.4
	128	16	156		-2.13006433	5.6e-6	+1.2
	160	32	310		-2.13004533	1.3e-5	-3.4
	192	64	618		-2.13005739	1.3e-6	
	224	128	1233		-2.13005868		

Table 3.4: Density function for the boundary curve (3.6.57), $\alpha = 10$ ($q_1 = 2$, $q_2 = 10$, $r = 10$ in (3.4.40)–(3.4.41)).

α	n	N	m	t	$u_{n,m}(t)$	error(t)	EOC
100	64	4	44	0	-1.90957749	9.6e-3	-1.2
	96	8	86		-1.91504627	4.1e-3	-1.9
	128	16	170		-1.91806231	1.1e-3	-1.8
	160	32	337		-1.91885787	3.3e-4	-2.3
	192	64	672		-1.91911880	6.7e-5	
	224	128	1342		-1.91918539		
	64	4	44	$\pi/3$	-2.02203770	2.3e-2	-9.6
	96	8	86		-1.99886217	3.0e-5	+5.9
	128	16	170		-1.99712031	1.8e-3	-2.3
	160	32	337		-1.99924289	3.5e-4	-6.0
	192	64	672		-1.99889806	5.5e-6	
	224	128	1342		-1.99889256		

Table 3.5: Density function for the boundary curve (3.6.57), $\alpha = 100$ ($q_1 = 2$, $q_2 = 100$, $r = 10$ in (3.4.40)–(3.4.41)).

integrand in (3.6.55) becomes more peaked as the point of evaluation approaches the boundary.

In Table 3.6 the "exact" solution is computed taking $n = 224$, $N = 128$, and $m = 1233$. In Table 3.7 the "exact" solution is computed taking $n = 224$, $N = 128$, and $m = 1342$. In each table the convergence rate of the method can be clearly seen, and is roughly $O(N^{-2})$ as expected.

3.7 Approximating the single layer potential

So far in §3.2–3.6, we have considered only second kind integral equations of the form (3.1.5) where the function f on the right hand side is known explicitly. In practice, for example if the Neumann problem for (3.1.1) is solved using Green's Third Identity (3.1.4) directly, the right hand side of (3.1.5) may be given by a single layer potential of the form (3.1.2), where v is the Neumann data. This can be written in parametric form as

$$(L_\alpha v)(t) := \frac{1}{2\pi} \int_{t-\pi}^{t+\pi} K_0(\alpha r(t,\tau))|\gamma'(\tau)|v(\tau)d\tau, \tag{3.7.59}$$

where K_0 is the modified Bessel function of order zero and r is given by (3.2.21). In this section we discuss how the quadrature method of §3.4 can be used to evaluate integrals of the form (3.7.59), and we also discuss how the error analysis of §3.5 may be affected if f is only known approximately.

Using the properties of K_0 (see Appendix B) it is clear that the kernel of $(L_\alpha v)(t)$ has a logarithmic singularity at $t = \tau$, and decays exponentially as $|\gamma(t) - \gamma(\tau)|$ becomes large. Thus as α increases, the kernel again becomes more and more peaked. Hence we use a similar method for the quadrature approximation of (3.7.59) as that used in §3.4 for the approximation of (3.2.22).

Making the transformation $\tau \to t - \tau/\alpha$ and following the same lines as in the derivation of (3.4.39) we get

$$\begin{aligned}(L_\alpha v)(t) \;=\; \frac{1}{2\pi\alpha} \int_0^{\alpha\pi} \Big\{ & K_0\left(\alpha r\left(t, t-\frac{\tau}{\alpha}\right)\right) \left|\gamma'\left(t-\frac{\tau}{\alpha}\right)\right| v\left(t-\frac{\tau}{\alpha}\right) \\ & + K_0\left(\alpha r\left(t, t+\frac{\tau}{\alpha}\right)\right) \left|\gamma'\left(t+\frac{\tau}{\alpha}\right)\right| v\left(t+\frac{\tau}{\alpha}\right) \Big\} d\tau.\end{aligned} \tag{3.7.60}$$

α	n	N	m	$\boldsymbol{x}$	$U_{n,m}(\boldsymbol{x})$	error($\boldsymbol{x}$)	EOC
10	64	4	41	(0,0)	1.09097301e-4	2.3e-7	-3.1
	96	8	79		1.08896358e-4	2.7e-8	-2.2
	128	16	156		1.08875238e-4	5.8e-9	-2.1
	160	32	310		1.08870788e-4	1.4e-9	-2.3
	192	64	618		1.08869687e-4	2.8e-10	
	224	128	1233		1.08869405e-4		
	64	4	41	(1,1)	1.08467931e-1	1.3e-4	-2.4
	96	8	79		1.08314618e-1	2.5e-5	-1.9
	128	16	156		1.08332630e-1	6.9e-6	-2.1
	160	32	310		1.08337877e-1	1.6e-6	-2.3
	192	64	618		1.08339195e-1	3.3e-7	
	224	128	1233		1.08339526e-1		

Table 3.6: PDE solution for the boundary curve (3.6.57), $\alpha = 10$ ($q_1 = 2$, $q_2 = 10$, $r = 10$ in (3.4.40)–(3.4.41)).

α	n	N	m	$\boldsymbol{x}$	$U_{n,m}(\boldsymbol{x})$	error($\boldsymbol{x}$)	EOC
100	64	4	44	(0,0)	7.13256905e-44	1.2e-44	-3.5
	96	8	86		6.05540698e-44	1.1e-45	-3.3
	128	16	170		5.95980125e-44	1.1e-46	-2.5
	160	32	337		5.95065827e-44	1.9e-47	-2.4
	192	64	672		5.94907935e-44	3.6e-48	
	224	128	1342		5.94872143e-44		
	64	4	44	(1,1)	3.51222936e-13	1.1e-13	-2.6
	96	8	86		2.62520077e-13	1.7e-14	-3.4
	128	16	170		2.47062494e-13	1.6e-15	-4.1
	160	32	337		2.45538423e-13	9.7e-17	-4.7
	192	64	672		2.45444904e-13	3.7e-18	
	224	128	1342		2.45441211e-13		
	64	4	44	(1.95,0)	8.85192110e-3	3.6e-5	-1.0
	96	8	86		8.79889776e-3	1.7e-5	-1.8
	128	16	170		8.81141689e-3	4.9e-6	-1.8
	160	32	337		8.81492525e-3	1.4e-6	-2.3
	192	64	672		8.81606404e-3	2.9e-7	
	224	128	1342		8.81635258e-3		

Table 3.7: PDE solution for the boundary curve (3.6.57), $\alpha = 100$ ($q_1 = 2$, $q_2 = 100$, $r = 10$ in (3.4.40)–(3.4.41)).

With the operator $\mathcal{Q}_m$ defined as in §3.4, we define the operator $\tilde{\mathcal{Q}}_m$ by truncation on the interval $[0, x_1)$ (where $x_1 = (1/N)^{q_1}$, see (3.4.40)), i.e.

$$\tilde{\mathcal{Q}}_m f(x) := \begin{cases} \mathcal{Q}_m f(x), & x \in [x_1, \infty) \\ 0, & x \in [0, x_1). \end{cases}$$

We then approximate $(L_\alpha v)(t)$ by

$$\begin{aligned}(L_{\alpha,m} v)(t) = \\ \frac{1}{2\pi\alpha} \int_0^{\alpha\pi} \tilde{\mathcal{Q}}_m \left\{ K_0\left(\alpha r\left(t, t - \frac{(\cdot)}{\alpha}\right)\right) \left|\gamma'\left(t - \frac{(\cdot)}{\alpha}\right)\right| \right\}(\tau) v\left(t - \frac{\tau}{\alpha}\right) \\ + \tilde{\mathcal{Q}}_m \left\{ K_0\left(\alpha r\left(t, t + \frac{(\cdot)}{\alpha}\right)\right) \left|\gamma'\left(t + \frac{(\cdot)}{\alpha}\right)\right| \right\}(\tau) v\left(t + \frac{\tau}{\alpha}\right) d\tau.\end{aligned} \tag{3.7.61}$$

In practice, we replace v by its trigonometric interpolating polynomial $\mathcal{P}_n v$ given by (3.1.12), and then as in §3.4, the integrals on the right hand side of (3.7.61) can easily be evaluated analytically. Note that as $\tilde{\mathcal{Q}}_m f = 0$ on $[0, x_1)$ we are only evaluating the integral (3.7.61) on a region bounded away from the logarithmic singularity.

The main difference between (3.7.61) and the approximation of (3.2.22) comes in the error analysis. In (3.7.61) we are applying the operator $\tilde{\mathcal{Q}}_m$ to the functions

$$g_{\alpha,t,\pm}(\tau) := K_0\left(\alpha r\left(t, t \pm \frac{\tau}{\alpha}\right)\right) \left|\gamma'\left(t \pm \frac{\tau}{\alpha}\right)\right|.$$

However, unlike the functions $f_{\alpha,t,\pm}$ of §3.4, these functions do not quite fit the assumptions of Theorem 3.4.2, because the logarithmic singularity of $K_0(x)$ is slightly worse than x^β for any $\beta \in (0, 1]$.

In order to prove the convergence of the quadrature approximation (3.7.61), we first need the following generalisation of the estimates in [63]. This concerns the application of the truncated trapezoidal rule to integrate $\log(x)$.

Lemma 3.7.1 *There exists a constant C independent of N such that*

$$\left| \int_0^1 \log(x) dx - \sum_{i=2}^{N} \frac{h_i}{2} \{\log(x_i) + \log(x_{i-1})\} \right| \leq C \frac{1}{N^2}, \tag{3.7.62}$$

where x_i, $i = 0, \ldots, N$ are the mesh points given by (3.4.40), with $q_1 > 2$, and $h_i = x_i - x_{i-1}$, $i = 1, \ldots, N$.

Proof: First define

$$\begin{aligned} E_N &:= \int_0^1 \log(x)dx - \sum_{i=2}^{N} \frac{h_i}{2}\{\log(x_i) + \log(x_{i-1})\} \\ &= \underbrace{\int_0^{x_1} \log(x)dx}_{I_1} + \underbrace{\int_{x_1}^1 \log(x)dx - \sum_{i=2}^{N} \frac{h_i}{2}\{\log(x_i) + \log(x_{i-1})\}}_{I_2}. \end{aligned} \tag{3.7.63}$$

Then we have

$$\begin{aligned} |I_1| &= |x_1 \log(x_1) - x_1| \\ &\leq \left|\left(\frac{1}{N}\right)^{q_1} \log\left(\left[\frac{1}{N}\right]^{q_1}\right)\right| + \left|\left(\frac{1}{N}\right)^{q_1}\right| \\ &\leq C\frac{1}{N^2}, \end{aligned} \tag{3.7.64}$$

provided $q_1 > 2$. We also have

$$\begin{aligned} |I_2| &= \left|\sum_{i=2}^{N} \left[\underbrace{\int_{x_{i-1}}^{x_i} \log(x)dx - \frac{h_i}{2}[\log(x_i) + \log(x_{i-1})]}_{E(\log x)_i}\right]\right| \\ &\leq \sum_{i=2}^{N} |E(\log x)_i| \\ &\leq C\sum_{i=2}^{N} h_i^2 \int_{x_{i-1}}^{x_i} \left|\frac{d^2}{dx^2}(\log(x))\right| dx \end{aligned}$$

where the last line follows by using Taylor's Theorem with integral remainder.

We then get

$$\begin{aligned} |I_2| &\leq C\sum_{i=2}^{N} h_i^2 \int_{x_{i-1}}^{x_i} \left|\frac{1}{x^2}\right| dx \\ &\leq C\sum_{i=2}^{N} h_i^3 \max_{[x_{i-1},x_i]} \left(\frac{1}{x^2}\right) \\ &= C\sum_{i=2}^{N} h_i^3 x_{i-1}^{-2} \\ &= C\sum_{i=2}^{N} h_i^3 \left(\frac{i-1}{N}\right)^{-2q_1}. \end{aligned}$$

Now we also have

$$\begin{aligned} h_i &= \left(\frac{i}{N}\right)^{q_1} - \left(\frac{i-1}{N}\right)^{q_1} \\ &= q_1 \left(\frac{1}{N}\right) (\xi)^{q_1-1}, \end{aligned}$$

for some $\xi \in (\frac{i-1}{N}, \frac{i}{N})$, by the Mean Value Theorem. This gives us

$$h_i \leq q_1 \left(\frac{1}{N}\right) \left(\frac{i}{N}\right)^{q_1-1}, \tag{3.7.65}$$

and so,

$$\begin{aligned} |I_2| &\leq C\sum_{i=2}^{N} \left(\frac{1}{N}\right)^3 \left(\frac{i}{N}\right)^{3q_1-3} \left(\frac{i-1}{N}\right)^{-2q_1} \\ &= C\sum_{i=2}^{N} \left(\frac{1}{N}\right)^3 \left(\frac{i}{N}\right)^{3q_1-3} \left(\frac{i}{N}\right)^{-2q_1} \underbrace{\left(\frac{i-1}{i}\right)^{-2q_1}}_{I_3}. \end{aligned}$$

Now as $i \in (2, N)$, $(i-1)/i$ is bounded above and below, hence I_3 is bounded by

a constant, and so

$$\begin{aligned} |I_2| &\leq C\sum_{i=2}^{N}\left(\frac{1}{N}\right)^3\left(\frac{i}{N}\right)^{q_1-3} \\ &\leq C\frac{1}{N^2}\left[\underbrace{\sum_{i=2}^{N}\frac{1}{N}\left(\frac{i}{N}\right)^{\epsilon-1}}_{I_4}\right], \end{aligned}$$

when $q_1 \geq 2+\epsilon$ for some $\epsilon \in (0,1)$. Now

$$I_4 \leq \int_{\frac{1}{N}}^{1} x^{\epsilon-1}dx \leq C,$$

and so

$$|I_2| \leq C\frac{1}{N^2}.$$

Combining the bounds on $|I_1|$ and $|I_2|$ we get the required result. □

We can now prove the main result of this section.

Theorem 3.7.2 *For all $v \in L_2$ there exists a constant C independent of α and m such that*

$$\|L_{\alpha,m}v - L_\alpha v\|_{L_\infty} \leq C\frac{1}{m^2}\|v\|_{L_2}.$$

Proof: By (3.7.60), (3.7.61)

$$\begin{aligned} \|L_{\alpha,m}v - L_\alpha v\|_{L_\infty} &= \sup_{t\in[0,2\pi]}\left|\frac{1}{2\pi\alpha}\int_0^{\alpha\pi}\left\{(\tilde{\mathcal{Q}}_m - I)\{g_{\alpha,t,-}(\cdot)\}(\tau)v\left(t-\frac{\tau}{\alpha}\right)\right.\right. \\ &\qquad \left.\left. + (\tilde{\mathcal{Q}}_m - I)\{g_{\alpha,t,+}(\cdot)\}(\tau)v\left(t+\frac{\tau}{\alpha}\right)\right\}d\tau\right|. \end{aligned}$$

For simplicity we consider only the first term on the right hand side.

$$\begin{aligned}
&\left|\int_0^{\alpha\pi}(\tilde{\mathcal{Q}}_m - I)\{g_{\alpha,t,-}(\cdot)\}(\tau)v\left(t-\frac{\tau}{\alpha}\right)\right| \\
&\leq \left|\int_0^{1}(\tilde{\mathcal{Q}}_m - I)\{g_{\alpha,t,-}(\cdot)\}(\tau)v\left(t-\frac{\tau}{\alpha}\right)\right| \\
&\quad + \left|\int_1^{\alpha\pi}(\tilde{\mathcal{Q}}_m - I)\{g_{\alpha,t,-}(\cdot)\}(\tau)v\left(t-\frac{\tau}{\alpha}\right)\right| \\
&=: I_1 + I_2.
\end{aligned}$$

It can be shown, using the same argument as for parts (d), (e) and (f) of the proof of Theorem 3.4.3 that $\|g_{\alpha,t,-} : e^{-\mu\tau}\|_2$ is bounded independently of α and t for all $\mu \in (0, C_1)$, where C_1 is the constant on the left hand side of (3.2.25). Hence on $[1, \alpha\pi]$ the error bound of Theorem 3.4.2 holds, and hence

$$I_2 \leq C\frac{1}{m^2}\|v\|_{L_2}$$

as in the proof of Theorem 3.4.4. An equivalent bound on I_1 follows from the ascending series representation for K_0 (see (B.0.3)), from which we can deduce that $K_0(z) = -\log z +$ smoother terms, and Lemma 3.7.1. □

The final question which we address in this section is that of how the error analysis of §3.5 is affected if the right hand side f of (3.1.6) involves a single layer potential which has to be discretised. Consider solving the integral equation

$$\lambda u + K_\alpha u(t) = L_\alpha g(t). \tag{3.7.66}$$

The fully discrete collocation method is then to find $u_{n,m} \in T_n$ satisfying

$$(\lambda I + K_{\alpha,m})u_{n,m}(t_j^{(n)}) = L_{\alpha,m}g(t_j^{(n)}), \quad j = 0, \ldots, 2n-1. \tag{3.7.67}$$

Theorem 3.7.3 *For n and m sufficiently large the discrete collocation equations (3.7.67) have a unique solution $u_{n,m} \in T_n$. The error estimate (3.5.53) holds, where u is the solution of (3.7.66).*

Proof: Existence and uniqueness of the solution follow directly from Theorem 3.5.5. However, following the same lines as the proof of Theorem 3.5.5 we

now have

$$u_{n,m} = (\lambda I + \mathcal{P}_n K_{\alpha,m})^{-1} \mathcal{P}_n L_{\alpha,m} g,$$

from which we get

$$u - u_{n,m} =$$
$$(\lambda I + \mathcal{P}_n K_{\alpha,m})^{-1} [\lambda(u - \mathcal{P}_n u) - \mathcal{P}_n(K_\alpha - K_{\alpha,m})u + \mathcal{P}_n(L_\alpha - L_{\alpha,m})g],$$

and the result follows from Theorem 3.7.2 and Theorem 3.5.5. □

3.8 First kind integral equations

In this section we shall discuss the solution of the first kind integral equations which can arise from (3.1.1). These are of the form

$$\mathcal{L}_\alpha u = f \quad \text{on } \Gamma, \tag{3.8.68}$$

with f a given function on Γ, and can arise when (3.1.1) is supplemented by Dirichlet data $U = g$ on Γ. If we choose to use Green's Third Identity (3.1.4) directly for this problem then we get (3.8.68) with $u = \partial U/\partial n$, and $f = \frac{1}{2}g + \mathcal{K}_\alpha g$. Alternatively using the indirect representation $U = \mathcal{L}_\alpha u$ for this problem leads again to (3.8.68), with $f = g$, the Dirichlet data. Here we will assume that f is known exactly. In practice, if $f = \frac{1}{2}g + \mathcal{K}_\alpha g$ then the integral $\mathcal{K}_\alpha g$ can be evaluated by replacing g by its trigonometric interpolating polynomial (3.1.12), and then using the quadrature scheme of §3.4.

Under the same assumptions on Γ as in §3.1, we can write (3.8.68) as

$$L_\alpha u(t) = f(t), \quad t \in [0, 2\pi], \tag{3.8.69}$$

with the operator L_α given by (3.7.59). We solve (3.8.69) using essentially the same collocation method as described in §3.1. We seek

$$u_n = \sum_{k=0}^{2n-1} a_k^{(n)} \phi_k \in T_n,$$

where the coefficients $\{a_k^{(n)}\}_{k=0}^{2n-1}$ are now defined by requiring that

$$L_\alpha u_n(t_j^{(n)}) = f(t_j^{(n)}), \quad j = 0, \ldots, 2n-1,$$

or equivalently

$$\sum_{k=0}^{2n-1} \left\{ L_\alpha \phi_k(t_j^{(n)}) \right\} a_k^{(n)} = f(t_j^{(n)}), \quad j = 0, \ldots, 2n-1.$$

We can evaluate the integrals $L_\alpha \phi_k(t_j^{(n)})$ as described in §3.7, and the fully discrete method is then

$$L_{\alpha,m} u_{n,m}(t_j^{(n)}) = f(t_j^{(n)}), \quad j = 0, \ldots, 2n-1. \tag{3.8.70}$$

The error analysis of this method is much harder than for the corresponding second kind integral equations. Indeed, it is our belief that it is not possible to derive a numerical method for solving (3.8.68) where the errors will be bounded in norm independently of α. It is shown in §3.3 that the operator $(\lambda I + K_\alpha)^{-1}$ exists and is bounded in $\|\cdot\|_{L_2}$ independently of α, and this was crucial for the error analysis in the second kind case. However, for the first kind equation (3.8.68) it is shown in [56, Theorem 2.1] that L_α is an isomorphism $L_\alpha : H^{-1/2}(\Gamma) \mapsto H^{1/2}(\Gamma)$ with

$$\|L_\alpha^{-1}\|_{H^{1/2} \mapsto H^{-1/2}} \leq C\alpha, \tag{3.8.71}$$

where the constant C is independent of α. If L_α^{-1} exists then from (3.8.69) we can write $u = L_\alpha^{-1} f$, but using (3.8.71), this implies that the magnitude of u may be of order α. If the solution blows up with respect to α then we would expect the errors to be of at least the same order with respect to α.

However, for fixed α we would still expect the same rate of convergence with respect to n and m as that acheived in Theorem 3.5.5 for the solution of the second kind integral equations, and even if the errors are blowing up with $\mathcal{O}(\alpha)$, this is still a lot better than the exponential blowup with respect to α when the splitting method (3.1.16) is used, as in [51].

We do not present an error analysis here for the solution of the first kind

integral equation (3.8.69), but rather we demonstrate the expected convergence properties of the method by means of two numerical examples.

First, we solve (3.8.69) with $f = 1$ for $\alpha = 10$ and $\alpha = 100$, using the method (3.8.70). This integral equation arises from solving (3.1.1) on a bounded domain $\Omega \subset \mathbb{R}^2$ with Dirichlet data $U = 1$ on the boundary, and then the solution U of (3.1.1) can be expressed as

$$U = \mathcal{L}_\alpha u. \tag{3.8.72}$$

Here we choose Ω to be the ellipse with boundary

$$\gamma(t) = (5\cos t, \sin t), \quad t \in [0, 2\pi].$$

We do not know the true solution of this problem, so in order to compute errors we compute a "near exact" solution using the equivalent second kind formulation (3.6.54) and (3.6.55). To do this we solve (3.6.54) using (3.5.52) with $n = 24$, $N = 320$, $m = 3079$ when $\alpha = 10$ or $m = 3352$ when $\alpha = 100$, and mesh parameters $q_1 = 2$, $q_2 = \alpha$ and $r = 10$. Note that this is the same example as in Tables 3.2–3.3, but whereas there we only solve for the density function u, here we also compute U using (3.6.55) at the points $(0, 0)$ and $(4, 0)$ using the trapezoidal rule with 384 nodes. The "near exact" solutions are shown in Table 3.8.

We then use (3.8.70) to solve (3.8.69) with $f = 1$ and compute (3.8.72) using the trapezoidal rule. In Tables 3.9 and 3.10 the approximate values of U are shown, for $\alpha = 10$ and $\alpha = 100$ respectively, and the errors are computed by treating the values in Table 3.8 as the exact solutions. Here the mesh parameters are chosen as $q_1 = 4$, $q_2 = \alpha$ and $r = 10$. As in §3.6 we double the number of quadrature points N on the interval $[0, 1]$, leading to approximately doubling of the total number of quadrature points m on the interval $[0, \alpha\pi]$, and then we increase the number of collocation points n logarithmically in N, ie. $n = C \log N$. We use $8n$ nodes for the trapezoidal rule evaluation of (3.8.72) when $\alpha = 10$, and $16n$ nodes when $\alpha = 100$. The estimated order of convergence is computed as in (3.6.58), and then as we expect the convergence to be superalgebraic in n and second order in N we predict that the entries in the EOC column should be approximately equal to two. This can be roughly seen in Tables 3.9 and 3.10 although the convergence is a little erratic. One explanation for this may be that

we are only using a "near exact" value for the true solution, rather than the exact true solution.

As a second example, we solve (3.8.69) with $f = \frac{1}{2}g + \mathcal{K}_\alpha g$, where

$$g = \frac{1}{2\pi} K_0(\alpha\sqrt{2.44 - 2.4\cos t}), \quad t \in [0, 2\pi],$$

and where Ω is the unit circle with boundary $\Gamma = (\cos t, \sin t)$. This integral equation arises from solving (3.1.1) using the direct boundary integral equation method, which corresponds to the middle line of (3.1.4). The Dirichlet data is given by $U|_\Gamma = g$, and then u represents the unknown Neumann data, $u = \partial U/\partial n$. The exact solution of (3.1.1) is then given by

$$U(\boldsymbol{x}) = \frac{1}{2\pi} K_0(\alpha|\boldsymbol{x} - (1.2, 0)|),$$

and then the Neumann data is

$$\frac{\partial U}{\partial n} = \frac{\alpha(1.2\cos t - 1)}{2\pi\sqrt{2.44 - 2.4\cos t}} K_1(\alpha\sqrt{2.44 - 2.4\cos t}), \quad t \in [0, 2\pi]. \tag{3.8.73}$$

The exact values of $\partial U/\partial n$ for $t = 0, \pi/4$ and $\alpha = 10, 100$ are shown in Table 3.11.

We solve (3.8.69) using (3.8.70), where the right hand side has been evaluated by replacing g by its trigonometric interpolating polynomial (3.1.12), and then using the quadrature scheme of §3.4. In Tables 3.12 and 3.13 the approximate values of $\partial U/\partial n$ at $t = 0, \pi/4$, and the errors between these values and the true solution (given in Table 3.11) are shown, for $\alpha = 10$ and $\alpha = 100$ respectively. Here the mesh parameters are chosen as $q_1 = 4$ on the left hand side, $q_1 = 2$ on the right hand side, $q_2 = \alpha$ and $r = 10$. To demonstrate the superalgebraic convergence rate with respect to n, we double n and for each fixed value of n we increase N starting from $N = 10$ until the errors will reduce no further, although we do not increase the value of N beyond $N = 1280$ because of the computational cost involved. Here m represents the number of quadrature points used on the left hand side of (3.8.70). The number of quadrature points used for the evaluation of the integrals on the right hand side is $m + 1$. The superalgebraic convergence rate with respect to n can clearly be seen for both $\alpha = 10$ and $\alpha = 100$, and these results appear to support the earlier assertion that this method is far more robust than the splitting method (3.1.16) when α becomes large.

α	$U(0,0)$	$U(4,0)$
10	9.26807277e-5	6.67617983e-3
100	7.59364551e-44	1.79860237e-25

Table 3.8: Near exact solutions computed using the second kind formulation (3.6.54).

α	n	N	m	$\boldsymbol{x}$	$U_{n,m}(\boldsymbol{x})$	error($\boldsymbol{x}$)	EOC
10	10	10	97	(0,0)	8.72588736e-05	5.4e-6	1.9
	12	20	193		9.12175601e-05	1.5e-6	1.9
	15	40	385		9.22853034e-05	4.0e-7	2.1
	18	80	770		9.25880528e-05	9.3e-8	3.0
	21	160	1539		9.26684365e-05	1.2e-8	0.9
	24	320	3078		9.26742648e-05	6.5e-9	
	10	10	97	(4,0)	6.39901384e-03	2.8e-4	1.9
	12	20	193		6.60342056e-03	7.3e-5	2.1
	15	40	385		6.65903577e-03	1.7e-5	1.6
	18	80	770		6.67055335e-03	5.6e-6	2.1
	21	160	1539		6.67491178e-03	1.3e-6	5.0
	24	320	3078		6.67613780e-03	4.2e-8	

Table 3.9: PDE solution computed using (3.8.70) and (3.8.72), $\alpha = 10$.

α	n	N	m	$\boldsymbol{x}$	$U_{n,m}(\boldsymbol{x})$	error($\boldsymbol{x}$)	EOC
100	10	10	106	(0,0)	7.19860942e-44	4.0e-45	1.7
	12	20	211		7.47353866e-44	1.2e-45	1.9
	15	40	420		7.56075385e-44	3.3e-46	2.0
	18	80	839		7.58531059e-44	8.3e-47	2.0
	21	160	1676		7.59150412e-44	2.1e-47	2.0
	24	320	3351		7.59311556e-44	5.3e-48	
	10	10	106	(4,0)	1.69508151e-25	1.0e-26	1.8
	12	20	211		1.77051694e-25	2.8e-27	2.0
	15	40	420		1.79151721e-25	7.1e-28	2.1
	18	80	839		1.79702289e-25	1.6e-28	2.9
	21	160	1676		1.79839347e-25	2.1e-29	0.8
	24	320	3351		1.79872606e-25	1.2e-29	

Table 3.10: PDE solution computed using (3.8.70) and (3.8.72), $\alpha = 100$.

α	$\partial U/\partial n$ $(t=0)$	$\partial U/\partial n$ $(t=\pi/4)$
10	2.22603465e-01	-2.24659711e-05
100	9.36317756e-09	-1.39706137e-38

Table 3.11: Exact values of $\partial U/\partial n$ given by (3.8.73).

α	n	N	m	t	$u_{n,m}(t)$	error(t)
10	4	10	97	0	1.82220861e-01	4.0e-2
	8	10	97		1.96151946e-01	2.6e-2
	16	10	97		2.20378688e-01	2.2e-3
	32	40	385		2.22875374e-01	2.7e-4
	64	1280	12309		2.22603820e-01	3.5e-7
	4	10	97	$\pi/4$	-3.37745011e-03	3.4e-3
	8	10	97		2.27427874e-03	2.3e-3
	16	10	97		6.40698117e-04	6.6e-4
	32	40	385		4.70354169e-06	2.7e-5
	64	1280	12309		-2.24159511e-05	5.0e-8

Table 3.12: Approximate values of $\partial U/\partial n$ and errors, $\alpha = 10$.

α	n	N	m	t	$u_{n,m}(t)$	error(t)
100	8	20	211	0	9.10160934e-09	2.6e-10
	16	20	211		9.13187873e-09	2.3e-10
	32	20	211		9.23920533e-09	1.2e-10
	64	20	211		9.36128911e-09	1.9e-12
	128	1280	13400		9.36319697e-09	1.9e-14
	8	20	211	$\pi/4$	1.61820316e-12	1.6e-12
	16	20	211		1.60312523e-12	1.6e-12
	32	20	211		1.34537246e-12	1.3e-12
	64	20	211		2.02990829e-13	2.0e-13
	128	1280	13400		4.35253184e-17	4.4e-17

Table 3.13: Approximate values of $\partial U/\partial n$ and errors, $\alpha = 100$.

Chapter 4

Evaluating the solution at many points

4.1 Introduction

In Chapter 3, we have described the use of boundary integral equation methods for solving the homogeneous Helmholtz type equation

$$-\Delta u(\boldsymbol{x}) + \alpha^2 u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega, \tag{4.1.1}$$

with given Dirichlet or Neumann data on the boundary. The direct method, corresponding to the middle line of (3.1.4), reformulates (4.1.1) as the boundary integral equation

$$\frac{1}{2}u(\boldsymbol{x}) + \mathcal{K}_\alpha u(\boldsymbol{x}) = \mathcal{L}_\alpha \frac{\partial u}{\partial n}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma, \tag{4.1.2}$$

where $\mathcal{L}_\alpha$ and $\mathcal{K}_\alpha$ are the single and double layer potential operators defined by (3.1.2) and (3.1.3) respectively. If either the Dirichlet data or the Neumann data is known, then (4.1.2) is a first or second kind integral equation respectively for the unknown complementary boundary data, and these integral equations can be solved using the collocation method of Chapter 3.

Having computed the complementary boundary data, $u(\boldsymbol{x})$ could in theory then be evaluated at any point $\boldsymbol{x} \in \Omega$ by using the top line of (3.1.4), which we

repeat here,

$$u(\boldsymbol{x}) = \mathcal{L}_\alpha \frac{\partial u}{\partial n}(\boldsymbol{x}) - \mathcal{K}_\alpha u(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \tag{4.1.3}$$

Now, in the context of the methods for solving inhomogeneous problems (1.0.1) discussed in Chapter 1, we will require the solution u of (4.1.1) at all the points in Ω at which we have computed the particular solution of (1.0.1), preferably to the same order of accuracy and with the same order cost. This is our goal in this chapter.

Recalling the accuracy and cost of the domain embedding method of Chapter 2, we thus wish to approximate the solution u of the boundary value problem for (4.1.1) at all the $\mathcal{O}(n^2)$ points of $R_h \cap \Omega$ (with the mesh R_h defined by (2.3.13)), to $\mathcal{O}(n^{-2})$ accuracy and with a cost no greater than $\mathcal{O}(n^2 \log n)$. Here n represents the number of grid points of R_h in each direction (see (2.3.13)).

In this section, we shall begin by discussing the cost and accuracy of solving (4.1.2) using the collocation method of Chapter 3. We demonstrate below that the cost of finding the complementary boundary data to an accuracy of $\mathcal{O}(n^{-p})$ is $\mathcal{O}(n^{\frac{p}{2}} \log^2 n)$, for any value of $p > 0$. We shall then discuss the problem of computing u at all points of $R_h \cap \Omega$ once the complementary boundary data has been computed. We shall explain why using the representation (4.1.3) at many points in Ω is not a good idea, and we will then discuss different approaches for finding $u(\boldsymbol{x})$ for many values of $\boldsymbol{x} \in \Omega$. Later (in §4.2) we will present a method for doing this (based on that in [59]) and we will show that choosing $p = 4$ above (ie. computing the complementary boundary data to $\mathcal{O}(n^{-4})$ accuracy) is sufficient to allow us to compute u at all points of $R_h \cap \Omega$ to $\mathcal{O}(n^{-2})$ accuracy and with a cost of $\mathcal{O}(n^2 \log^2 n)$ as required.

So, first recall that we have proved (in Theorem 3.5.5 for the second kind integral equations arising from (4.1.2) and experimentally in §3.8 for the first kind integral equations arising from (4.1.2)) that the collocation method is superalgebraically accurate in terms of the number of collocation points (which we denote here by n_c), and $\mathcal{O}(m^{-2})$ accurate in terms of the number of quadrature points m. In order to evaluate the complementary boundary data to $\mathcal{O}(n^{-p})$ accuracy, we would then need to choose $n_c = \mathcal{O}(p \log n)$, and $m = \mathcal{O}(n^{\frac{p}{2}})$. We now consider the cost of this.

The collocation method consists of solving a $2n_c \times 2n_c$ linear system, either (3.5.52) for the second kind integral equation or (3.8.70) for the first kind integral equation. To set up these linear systems requires the computation of each matrix element using the quadrature scheme of §3.4. This consists of evaluating an integral using the trapezoidal rule (on a graded mesh) with m quadrature points, and so the cost for the evaluation of each matrix element is $\mathcal{O}(m)$. Thus the total cost to set up each linear system is $\mathcal{O}(mn_c^2)$.

The solution of each linear system requires the inversion of a full $2n_c \times 2n_c$ matrix. The cost of this solve is $\mathcal{O}(n_c^3)$.

So, taking $n_c = \mathcal{O}(p \log n)$ and $m = \mathcal{O}(n^{\frac{p}{2}})$, in order to evaluate the boundary data to $\mathcal{O}(n^{-p})$ accuracy as described above, the cost of setting up the linear system will be $\mathcal{O}(n^{\frac{p}{2}} \log^2 n)$, and the cost of the solve will be $\mathcal{O}(\log^3 n)$. To keep these costs to roughly the same order as the cost of computing the particular solution (namely $\mathcal{O}(n^2 \log n)$) we can then afford to choose $p = 4$, ie. we can evaluate the complementary boundary data to $\mathcal{O}(n^{-4})$ accuracy at a cost of $\mathcal{O}(n^2 \log^2 n)$. Alternatively, choosing $p = 2$ or $p = 3$ would allow us to evaluate the complementary boundary data to $\mathcal{O}(n^{-2})$ or $\mathcal{O}(n^{-3})$ accuracy at a cost lower than that of computing the particular solution. We discuss our accuracy requirements further, and determine exactly what value of p we must choose, in §4.2.

The question now is how to compute u at all points of $R_h \cap \Omega$ once the complementary boundary data is known (to a certain order of accuracy). First, we shall consider the cost and accuracy of computing (4.1.3) directly at every point $\boldsymbol{x} \in R_h \cap \Omega$.

Under the assumptions of Chapter 3, namely that γ satisfies (1.1.5), $\partial u/\partial n|_\Gamma$ and $u|_\Gamma$ will both be smooth 2π-periodic functions. If we evaluate the integrals on the right hand side of (4.1.3) using the trapezoidal rule on a uniform boundary grid, the asymptotic convergence rate will then be superalgebraic (this can be proved using the Euler-Maclaurin summation formulae, see [42, Theorem 6.2] for details). We could thus *in principle* achieve the same order of accuracy as for the collocation method (namely $\mathcal{O}(n^{-p})$) by evaluating (4.1.3) using the trapezoidal rule with the collocation points from the solution of (4.1.2) as the nodes. With $n_c = \mathcal{O}(p \log n)$, u could then *in principle* be evaluated at the $\mathcal{O}(n^2)$ mesh points in $R_h \cap \Omega$ to $\mathcal{O}(n^{-p})$ accuracy at a cost of $\mathcal{O}(n^2 \log n)$.

However, as we shall show in §4.3, the kernels of $\mathcal{L}_\alpha$ and $\mathcal{K}_\alpha$ blow up like $\log \epsilon$

and ϵ^{-1} respectively as the point of evaluation approaches the boundary, where ϵ is the distance from Γ. As a result, when $\boldsymbol{x}$ is close to Γ, the accuracy acheived by the direct numerical evaluation of (4.1.3) can be very poor. We discuss this further, and suggest an alternative approach for evaluating $u(\boldsymbol{x})$ when $\boldsymbol{x}$ is near Γ, in §4.3.

What this means in practice is that as the number of mesh points in R_h increases, and new mesh points closer and closer to Γ appear, an increased number of nodes are needed for the trapezoidal rule evaluation of (4.1.3). So, although *in principle* the cost is only $\mathcal{O}(n^2 \log n)$, *in practice* the constants involved in this cost estimate may be very large. We present a numerical example demonstrating this in §4.4.

The cost of evaluating (4.1.3) directly at many points is considered to be one of the main drawbacks of using the boundary integral equation method for solving problems such as (4.1.1) when the solution is sought throughout Ω, and so in this chapter we seek an alternative approach.

One method for speeding up the evaluation of (4.1.3) is the Fast Multipole Method (FMM), which first appeared in the context of integral equations in Rokhlin [75]. In general the FMM can be thought of as a method for approximating certain types of matrix vector multiplication in such a way that acceptable accuracy can be obtained but with substantially less arithmetical work. For example, if we approximate

$$v(\boldsymbol{x}) = \int_\Gamma K(\boldsymbol{x}, \boldsymbol{y}) w(\boldsymbol{y}) d\boldsymbol{y}$$

for N values of $\boldsymbol{x}$ by a sum of the form

$$v_N(\boldsymbol{x}) = \sum_{i=1}^{N} K(\boldsymbol{x}, \boldsymbol{y}_i) W_i,$$

where $W_i = \mu_i w(\boldsymbol{y}_i)$ and μ_i are some appropriate weights, then

$$\boldsymbol{v} = K\boldsymbol{W},$$

where $\boldsymbol{v} = (v_N(\boldsymbol{x}_1), \ldots, v_N(\boldsymbol{x}_N))^T$, $\boldsymbol{W} = (W_1, \ldots, W_N)^T$, and K is the matrix whose (i,j)th entry is $K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{y}_j)$. Using the FMM could then reduce the

cost of evaluating KW from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log^p N)$, for some small positive constant p.

The FMM has been applied to the integrals arising from the solution of the two-dimensional Helmholtz equation with real wavenumber in Rokhlin [76], and its use for speeding up the evaluation of integrals of the form (4.1.3) is also suggested in Mckenney, Greengard and Mayo [62]. For a full description of the method, and a complete literature review, we refer to Beetson and Greengard [7].

Here we use a different approach, for two reasons.

i). Although it is true that the FMM can be used to speed up the numerical evaluation of (4.1.3), as we discussed above (and shall demonstrate in §4.3) the accuracy acheived by direct numerical evaluation of (4.1.3) can be very poor when $\boldsymbol{x}$ is near Γ, and the FMM does nothing to improve this. It is really preferable to avoid numerical integration altogether when $\boldsymbol{x}$ is near Γ.

ii). As we have used a domain embedding approach already in Chapter 2 (for finding a particular solution to the inhomogeneous problem (1.0.1)), it makes sense to take advantage of the same domain embedding framework here.

The method we will actually use to compute u throughout $R_h \cap \Omega$ is similar to that developed by Mayo in [59] for finding the solution to Laplace's equation at many points. It uses a similar domain embedding approach to that of Chapter 2, and has much in common with the idea there for evaluating the domain integral quickly at many points. As explained in Chapter 2, the main idea behind the method is not to evaluate the integrals directly at every point, but rather to solve an inhomogeneous PDE on the extended domain, using a fast solver. We describe the method further in §4.2.

The key requirement of the whole scheme however is the evaluation of $u(\boldsymbol{x})$ explicitly at a small number of points near Γ. As mentioned above the direct evaluation of (4.1.3) at points near Γ is problematic, and this is a difficulty which we deal with here.

In [59], Mayo's approach to this difficulty is to evaluate (4.1.3) directly by the trapezoidal rule, using a greater number of node points when $\boldsymbol{x}$ is near Γ. In §4.3 we show how this approach rapidly becomes unfeasible, and we present a new method for evaluating the solution near the boundary. This method, hereafter

referred to as the *extraction method* because of the way it uses an "extraction" technique to pull the solution off the boundary, is similar in some ways to a method developed by Wendland et. al. in [77], [78], but there are some crucial differences in our approach. We discuss this further in §4.3.

The extraction method avoids altogether the need for the direct evaluation of any integrals of the form (4.1.3). It can however only be applied when (4.1.1) is solved by the direct boundary integral equation method (4.1.2), and so in §4.3 we also present some alternative ideas for evaluating the integrals in (4.1.3) directly, although these approaches are less well-developed.

Finally in §4.4 we show how the method can be implemented in practice, and we discuss the details of the computational difficulties that arise. Numerical results demonstrating the cost and accuracy of the method are also presented.

4.2 Mayo's method for evaluating the solution at many points

As a first step, we embed Ω in the square R, given by (2.3.10), and we cover R in the uniform mesh R_h (2.3.13). The basic idea is then to evaluate $u(\boldsymbol{x})$, the solution of (4.1.1), explicitly at only a small number of carefully chosen mesh points near Γ, and then to use a fast Helmholtz solver to extend the solution to the rest of Ω.

First, we define the mesh function

$$U_{i,j} = \begin{cases} u(x_i, y_j), & (x_i, y_j) \in R_h \cap \Omega \\ 0, & (x_i, y_j) \in R_h \cap R \setminus \Omega, \end{cases} \tag{4.2.4}$$

where u is the solution of (4.1.1) and R_h is given by (2.3.13). For simplicity we assume no mesh points lie on Γ. Now with h denoting the mesh width, and using Lemma 2.3.3, it is clear that at regular mesh points inside Ω,

$$\begin{aligned} -\Delta_h U_{i,j} + \alpha^2 U_{i,j} &= -\Delta u + \alpha^2 u + \mathcal{O}(h^2) \\ &= 0 + \mathcal{O}(h^2), \end{aligned} \tag{4.2.5}$$

and at regular mesh points outside Ω we have

$$-\Delta_h U_{i,j} + \alpha^2 U_{i,j} = 0 \tag{4.2.6}$$

exactly. At irregular mesh points we define

$$\begin{aligned} f_{i,j} &:= -\Delta_h U_{i,j} + \alpha^2 U_{i,j} \\ &= \frac{1}{h^2}\left[-U_{i,j+1} - U_{i,j-1} - U_{i+1,j} - U_{i-1,j} + (4+\alpha^2 h^2)U_{i,j}\right]. \end{aligned} \tag{4.2.7}$$

Now let $B1$ denote the set of irregular mesh points in Ω, and define $B2$ to be the set of regular mesh points in Ω with at least one nearest neighbour (see Definition 2.3.1) lying in $B1$. Then to evaluate an approximation $\tilde{f}_{i,j}$ to $f_{i,j}$ we just need to evaluate $u(\boldsymbol{x})$ explicitly on $B1 \cup B2$. Assuming we have done this (using some yet to be determined method), we can then solve the system

$$-\Delta_h \tilde{U}_{i,j} + \alpha^2 \tilde{U}_{i,j} = \begin{cases} 0, & (x_i, y_j) \text{ regular} \\ \tilde{f}_{i,j}, & (x_i, y_j) \text{ irregular,} \end{cases} \tag{4.2.8}$$

$$\tilde{U}_{i,j} = 0, \quad (x_i, y_j) \text{ on } \partial R, \tag{4.2.9}$$

using the Fast Sine Transform method of §2.4.

We now need to consider the accuracy and cost of this method. First, we discuss the cost.

Using this method, we can compute an appropriate approximation to $u(\boldsymbol{x})$ at every mesh point in $R_h \cap \Omega$ using evaluations only at the $\mathcal{O}(n)$ points of $B1 \cup B2$. If $u(\boldsymbol{x})$ was evaluated at these points by evaluating (4.1.3) directly using the trapezoidal rule with $\mathcal{O}(n_c)$ nodes, then with $n_c = \mathcal{O}(\log n)$ as described in §4.1 the combined cost of all these evaluations would be $\mathcal{O}(n \log n)$. The largest asymptotic cost would then be that of solving (4.2.8)–(4.2.9). As described in §2.4, the cost of this solve is $\mathcal{O}(n^2 \log n)$. Although asymptotically this is the same as if the integrals (4.1.3) were evaluated directly by the trapezoidal rule at every point of Ω, the method using the fast Helmholtz solver is much faster, since the asymptotic constants are much smaller. Numerical experiments demonstrating this can be seen in §4.4. In practice we use the extraction method to evaluate u on $B1 \cup B2$, and we discuss the cost of this in §4.3.

We now consider the accuracy of this method. Using the error bound (2.3.22),

it is clear that if we can evaluate $\tilde{f}_{i,j}$ in such a way that it is an $\mathcal{O}(h^2)$ accurate approximation to $f_{i,j}$, (recalling that $h = \mathcal{O}(n^{-1})$) then we will have an error bound of the form

$$\max_{(x_i,y_j)\in R_h\cap\Omega} |\tilde{U}_{i,j} - u(x_i, y_j)| \leq Ch^2.$$

Now the formula (4.2.7) appears to indicate that in order to compute an $\mathcal{O}(h^2)$ accurate approximation to $f_{i,j}$ we would need to compute an $\mathcal{O}(h^4)$ accurate approximation to $u(x_i, y_j)$ on $B1 \cup B2$. This is because the evaluation of $f_{i,j}$ involves dividing the values of u on $B1 \cup B2$ by h^2.

However, this should not be a problem, because as described in §4.1, we can use the collocation method of Chapter 3 to compute $\partial u/\partial n$ and $u|_\Gamma$ to $\mathcal{O}(h^4)$ accuracy at a cost of $\mathcal{O}(n^2 \log^2 n)$, and we shall demonstrate in §4.3 that this is sufficient to enable us to then evaluate u to $\mathcal{O}(h^4)$ accuracy on $B1 \cup B2$. Note also that Mayo claims in [59] that in practice it is sufficient to evaluate u to $\mathcal{O}(h^3)$ or even $\mathcal{O}(h^2)$ accuracy on $B1 \cup B2$ in order to maintain the overall $\mathcal{O}(h^2)$ accuracy of the method. We present a numerical experiment to test this theory in §4.4.

So, provided the solution $u(\boldsymbol{x})$ of (4.1.1) can be found to sufficient accuracy near Γ, we can then achieve roughly the same orders of cost and accuracy as we did in Chapter 2 for the evaluation of the particular solution. The problem with this is that if (4.1.3) is used to evaluate u then the integrals in (4.1.3) are at their most difficult when $\boldsymbol{x}$ is near Γ. The extraction method proposed below, on the other hand, has the property that it works *best* when $\boldsymbol{x}$ is near Γ. This is discussed further in §4.3.

Before moving on to §4.3 we briefly mention an alternative approach which does not require the evaluation of u at any points near Γ. This method is applied in [58] to the solution of Laplace's equation, and here we describe its application to the solution of (4.1.1), supplemented by Dirichlet data $u|_\Gamma = f$, using an indirect boundary integral equation method. First, we note that the double layer potential $\mathcal{K}_\alpha\mu$ with continuous density μ satisfies (4.1.1), not just in Ω, but in all of $\mathbb{R}^2 \setminus \Gamma$ (although it is not continuous across Γ). It also satisfies the Dirichlet boundary condition provided μ satisfies

$$-\frac{1}{2}\mu(\boldsymbol{x}) + \mathcal{K}_\alpha\mu(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma. \tag{4.2.10}$$

We can solve this for μ using the collocation method (3.5.52) of Chapter 3, and then define the mesh function

$$w_{i,j} = (\mathcal{K}_\alpha \mu)(x_i, y_j), \quad (x_i, y_j) \in R \setminus \Gamma. \tag{4.2.11}$$

Analogously to (4.2.4), we again have

$$\begin{aligned} -\Delta_h w_{i,j} + \alpha^2 w_{i,j} &= (-\Delta + \alpha^2)\mathcal{K}_\alpha \mu + \mathcal{O}(h^2) \\ &= 0 + \mathcal{O}(h^2), \end{aligned}$$

at all regular mesh points in R. If we could then find

i). The values of $-\Delta_h w_{i,j} + \alpha^2 w_{i,j}$ at the irregular points,

ii). The values of $\mathcal{K}_\alpha \mu(x_i, y_j)$ on ∂R,

then we would have a linear system similar to (4.2.8)–(4.2.9) for $w_{i,j}$, and we could again solve this using the Fast Sine Transform method.

The key point about this alternative approach is that instead of evaluating $-\Delta_h w_{i,j} + \alpha^2 w_{i,j}$ directly at the irregular points, we can use the definition of $w_{i,j}$ (4.2.11) to derive an $\mathcal{O}(h^2)$ accurate formula for the values of $-\Delta_h w_{i,j} + \alpha^2 w_{i,j}$ at the irregular points in terms of the jump discontinuites in $\mathcal{K}_\alpha \mu(\boldsymbol{x})$ and its derivatives across Γ. This method is similar to that in §2.3.2 for computing the jump discontinuities of the derivatives of the domain integral (2.1.6), and for details we refer to [58].

The remaining problem of computing the values of $\mathcal{K}_\alpha \mu(x_i, y_j)$ on ∂R can be solved by using straightforward numerical integration (eg. the trapezoidal rule). Provided ∂R is chosen to be well separated from Γ there is no problem with the kernels of the integrals blowing up.

In terms of accuracy and complexity there is little to choose between the methods of [59] and [58]. Both methods have been used in the literature. For example, the method of [58] is used in [61], and an error analysis applicable to the method of [58] and carried out in a Besov space setting can be found in [48]. The method of [59] is preferred however in [62] and [29].

It is our opinion that in the absence of an accurate method for evaluating u near Γ, the method of [58] would be preferable. However, having developed the

extraction method (described in §4.3 below) for evaluating u near Γ we prefer to use the method of [59], as this allows the use of a direct boundary integral equation method for the solution of (4.1.1), which also provides us with the unknown boundary data, as well as providing a unified framework for the solution of both the Dirichlet and Neumann problems. The direct approach is also more standard in the engineering literature.

4.3 Evaluating the solution near the boundary

To implement the method of §4.2 we need a method for evaluating the solution u of (4.1.1) accurately near Γ. In this section we shall first present some ideas for evaluating (4.1.3) directly, and after having identified some problems with these methods we shall describe the *extraction method*, which is the method we will use in practice.

4.3.1 Evaluating (4.1.3) directly

To evaluate the integrals of (4.1.3) explicitly, for $\boldsymbol{x}$ near Γ, we need to evaluate

$$\mathcal{L}_\alpha \frac{\partial u}{\partial n}(\boldsymbol{x}) = \int_\Gamma \Phi_\alpha(\boldsymbol{x},\boldsymbol{y}) \frac{\partial u}{\partial n}(\boldsymbol{y}) d\boldsymbol{y}, \quad \text{and} \quad \mathcal{K}_\alpha u(\boldsymbol{x}) = \int_\Gamma \frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial n(\boldsymbol{y})} u(\boldsymbol{y}) d\boldsymbol{y},$$

where Φ_α is the fundamental solution of the Helmholtz equation, given by (2.1.5). If $\boldsymbol{x} \notin \Gamma$ and $\boldsymbol{y} \in \Gamma$, then it can be easily shown (see Appendix B) that

$$\Phi_\alpha(\boldsymbol{x},\boldsymbol{y}) = \mathcal{O}(-\log|\boldsymbol{x}-\boldsymbol{y}|), \quad \text{as } \boldsymbol{x} \to \boldsymbol{y},$$

and so if the point of evaluation $\boldsymbol{x}$ is a distance ϵ from Γ then the kernel of $\mathcal{L}_\alpha \frac{\partial u}{\partial n}(\boldsymbol{x})$ is highly peaked near $\boldsymbol{x}$, with the peak being of height $\mathcal{O}(-\log\epsilon)$, as $\epsilon \to 0$.

The double layer potential $\mathcal{K}_\alpha u(\boldsymbol{x})$ is even more highly peaked for $\boldsymbol{x}$ near Γ. To see this precisely some care is needed. First note that when $\boldsymbol{x}$, $\boldsymbol{y}$ are both on Γ, the smoothness of Γ ensures that

$$\frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial n(\boldsymbol{y})} = \mathcal{O}(|\boldsymbol{x}-\boldsymbol{y}|\log|\boldsymbol{x}-\boldsymbol{y}|), \quad \text{as } \boldsymbol{x} \to \boldsymbol{y},$$

ie. the double layer potential as an operator on Γ has a very mild singularity. For fixed $\boldsymbol{x} \notin \Gamma$, the kernels of $\mathcal{L}_\alpha \frac{\partial u}{\partial n}(\boldsymbol{x})$ and $\mathcal{K}_\alpha u(\boldsymbol{x})$ are smooth and 2π-periodic. However, if $\boldsymbol{x} \notin \Gamma$ and $\boldsymbol{y} \in \Gamma$ then it can be shown by elementary calculus that

$$\frac{\partial \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})}{\partial n(\boldsymbol{y})} = \mathcal{O}(|\boldsymbol{x} - \boldsymbol{y}|^{-1}), \quad \text{as } \boldsymbol{x} \to \boldsymbol{y}. \tag{4.3.12}$$

Thus if we evaluate (4.1.3) using the trapezoidal rule on a uniform boundary grid, then although the asymptotic convergence rate is superalgebraic, the asymptotic constant in the superalgebraic error estimate will blow up as $\boldsymbol{x} \to \Gamma$.

More precisely, recall that if we use the trapezoidal rule with N nodes to approximate the integral $\int_0^{2\pi} f(x)dx$, where f is a smooth, 2π-periodic function, we get an error bound of the form

$$\left| \int_0^{2\pi} f(x)dx - \frac{2\pi}{N} \sum_{j=0}^{N-1} f\left(\frac{2\pi j}{N}\right) \right| \leq C \frac{1}{N^r} \|f\|_{r,\infty}, \quad \text{for all } r \in \mathbb{N}.$$

So, the asymptotic convergence rate is superalgebraic, however the asymptotic constant involves $\|f\|_{r,\infty}$. Thus if we attempt to approximate $\mathcal{K}_\alpha u(\boldsymbol{x})$ using the trapezoidal rule for $\boldsymbol{x}$ a distance ϵ from Γ, then the asymptotic constant in the error estimate will be a function of ϵ — we conjecture $\mathcal{O}(\epsilon^{-(r+1)})$ in this case, in view of the estimate (4.3.12).

Supposing this was the case, the error bound for the approximation of $\mathcal{K}_\alpha u(\boldsymbol{x})$ by the trapezoidal rule would then look like

$$\begin{aligned} |\mathcal{K}_\alpha u - T_N\{\mathcal{K}_\alpha u\}| &\leq C \frac{1}{N^r} \frac{1}{\epsilon^{r+1}}, \\ &= \frac{C}{\epsilon} \frac{1}{(N\epsilon)^r}. \end{aligned} \tag{4.3.13}$$

When ϵ is not small, the trapezoidal rule with equally spaced nodes will be very accurate, but when ϵ is small, the asymptotic constant C/ϵ will be large. Also, the superalgebraic convergence will not be seen at all until

$$N > 1/\epsilon. \tag{4.3.14}$$

This is borne out by the following numerical experiment.

Example 4.3.1 *In this experiment, we evaluate the solution of (4.1.1), where* Ω *is the unit circle with boundary* $\gamma(t) = (\cos t, \sin t)$ *and* $\alpha = 1$. *We solve the problem with exact solution*

$$u(\boldsymbol{x}) = \frac{1}{2\pi} K_0(\alpha|\boldsymbol{x} - (2,0)|), \tag{4.3.15}$$

in which case the boundary data is given by

$$\begin{aligned} u|_\Gamma &= \frac{1}{2\pi} K_0(\alpha\sqrt{5 - 4\cos t}), \quad t \in [0, 2\pi] \\ \frac{\partial u}{\partial n} &= \frac{\alpha(2\cos t - 1)}{2\pi\sqrt{5 - 4\cos t}} K_1(\alpha\sqrt{5 - 4\cos t}), \quad t \in [0, 2\pi]. \end{aligned}$$

We then evaluate u *at the point* $\boldsymbol{x} = (1-\epsilon, 0)$ *using the formula (4.1.3), where the integrals are evaluated using the trapezoidal rule with a uniform boundary mesh given by*

$$t_i = \frac{2\pi i}{N}, \quad i = 0, \ldots, N-1.$$

The errors between the approximate and exact values of $u(1-\epsilon, 0)$ *for various values of* ϵ *are given in Table 4.1. These numerical results clearly follow the*

N	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$	$\epsilon = 0.0001$
8	6.0e-2	8.3e-1	8.4e+0	8.4e+1
16	1.7e-2	4.0e-1	4.2e+0	4.2e+1
32	2.5e-3	1.8e-1	2.1e+0	2.1e+1
64	8.2e-5	7.5e-2	1.0e+0	1.0e+1
128	9.6e-8	2.6e-2	4.9e-1	5.2e+0
256	1.3e-13	5.6e-3	2.3e-1	2.6e+0
512	4.9e-17	3.9e-4	1.0e-1	1.3e+0
1024		2.3e-6	3.8e-2	6.2e-1

Table 4.1: Errors evaluating (4.1.3) at $(1-\epsilon, 0)$ using trapezoidal rule, uniform mesh.

pattern predicted by (4.3.13). When ϵ *is small the convergence is very slow up to a point, and then very rapid once* N *is large enough. Note that as* ϵ *changes from 0.1 to 0.01 the value of* N *needed to ensure an accuracy of* 6×10^{-2} *changes from*

$N = 8$ *to about* $N < 128$ *which resonates with the expected estimate (4.3.14).*

In practice, the convergence is not rapid enough for a small enough value of N to make this a feasible way of evaluating the integrals and an alternative approach is needed.

In [59] it is proposed that (4.1.3) be evaluated using the trapezoidal rule with equally spaced nodes, as described above, but rather than evaluating (4.1.3) directly on $B1$, it is suggested that the integrals be evaluated only on $B2$, or even $B3$, the set of regular points in Ω with at least one nearest neighbour lying in $B2$. The solution on $B1$ can then be determined by interpolation. However, although $B2$ and $B3$ lie further from Γ than $B1$, as the mesh width h decreases, the points of $B2$ and $B3$ will still approach Γ, and hence the errors will still blow up sooner or later. This blow up in the errors as $h \to 0$ is alluded to in [59], but there it is suggested that one merely use an increasing number of boundary points to deal with this. As demonstrated in Example 4.3.1, this approach will be insufficient when h becomes very small.

An alternative approach to using the trapezoidal rule with equally spaced nodes is to use a graded mesh scheme, ie. the composite trapezoidal rule with the mesh points bunched near where the kernel of the integrand varies quickly. We have experimented with various meshes, and the following example demonstrates how the results of Example 4.3.1 can be improved upon.

Example 4.3.2 *We solve exactly the same problem as in Example 4.3.1, but instead of using equally spaced mesh points we use the graded mesh, with mesh points given by*

$$t_i = \pm\pi \left(\frac{i}{N}\right)^q, \quad i = -N, \ldots, N. \tag{4.3.16}$$

The optimal value of q *was determined experimentally, and varies depending on* ϵ. *The errors between the approximate and exact values of* $u(1-\epsilon, 0)$ *for various values of* ϵ *are given in Table 4.2. As the mesh widths are no longer uniform, the superalgebraic convergence in terms of* N *is replaced by a convergence rate of* $\mathcal{O}(N^{-2})$. *This can be clearly seen in Table 4.2, and the errors are essentially independent of* ϵ.

N	$\epsilon = 0.1$ $q = 2$	$\epsilon = 0.01$ $q = 4$	$\epsilon = 0.001$ $q = 6$	$\epsilon = 0.0001$ $q = 8$
8	4.9e-3	1.7e-2	4.7e-2	9.6e-2
16	1.2e-3	4.1e-3	9.2e-3	1.6e-2
32	2.9e-4	1.0e-3	2.2e-3	3.9e-3
64	7.3e-5	2.6e-4	5.5e-4	9.6e-4
128	1.8e-5	6.4e-5	1.4e-4	2.4e-4
256	4.5e-6	1.6e-5	3.4e-5	6.0e-5
512	1.1e-6	4.0e-6	8.6e-6	1.5e-5

Table 4.2: Errors evaluating (4.1.3) at $(1-\epsilon, 0)$ using trapezoidal rule, graded mesh.

Comparing the two examples, it is clear that if N was taken to be large enough, the uniform grid would eventually be better for any value of ϵ, because of the superalgebraic convergence rate. Our experiments demonstrate this clearly for $\epsilon = 0.1$, and also less clearly for $\epsilon = 0.01$. However, in practice when ϵ is small the non-uniform grid gives much better results.

The problem with using a non-uniform grid in the overall context of the method of §4.2 is that the optimal mesh parameters have been determined merely by experimentation. At any particular point of evaluation, this method can be adapted to get good convergence rates, but despite some work in the literature on the problem of developing error estimates for the product integration of layer potentials using graded meshes similar to (4.3.16), see for example [11], the question of developing an explicit formula for the parameter q in (4.3.16) in terms of the distance to the boundary ϵ has not been fully answered. Another difficulty is that it may be necessary to devise different meshes on Γ for each $\boldsymbol{x}$ near Γ, depending on exactly how close $\boldsymbol{x}$ is to Γ, in which case this method could become very expensive. Hence we need an alternative approach.

One method that has been suggested in the literature for the problem of evaluating the double layer potential integrals $\mathcal{K}_\alpha$ in the case of Laplace's equation ($\alpha = 0$) is *singularity subtraction* (see eg. [83, p.55] or [28, §2.3]). If $\boldsymbol{y}_0$ is the

closest point on Γ to $\boldsymbol{x}$ then we can write

$$\int_\Gamma \frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial n(\boldsymbol{y})} u(\boldsymbol{y}) d\boldsymbol{y} = \int_\Gamma \frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial n(\boldsymbol{y})} [u(\boldsymbol{y}) - u(\boldsymbol{y}_0)] d\boldsymbol{y} + u(\boldsymbol{y}_0) \int_\Gamma \frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial n(\boldsymbol{y})} d\boldsymbol{y}. \quad (4.3.17)$$

The first integral on the right hand side will then be less peaked, and hence easier to evaluate numerically than the integral arising in the direct evaluation of $\mathcal{K}_\alpha u(\boldsymbol{x})$. If we were solving Laplace's equation, with fundamental solution $\Phi_L(\boldsymbol{x},\boldsymbol{y}) = -(1/2\pi)\log|\boldsymbol{x}-\boldsymbol{y}|$, then the second integral would be known analytically. As the principal singularities of the fundamental solutions to Laplace's equation and Helmholtz's equation are the same, we can write

$$\Phi_\alpha(\boldsymbol{x},\boldsymbol{y}) = \Phi_L(\boldsymbol{x},\boldsymbol{y}) + \{\text{smoother terms}\}.$$

Subtracting $\partial\Phi_L(\boldsymbol{x},\boldsymbol{y})/\partial n(\boldsymbol{y})$ out of the kernel of the second integral in (4.3.17) leaves a smoother integrand which can again be evaluated more easily numerically, plus a part which is known analytically.

However this method does not apply so readily to the single layer potential $\mathcal{L}_\alpha$ which also appears in (4.1.3) since there is no analytic form for $\int_\Gamma \Phi_L(\boldsymbol{x},\boldsymbol{y})$. Thus singularity subtraction does not provide a generally applicable answer to the problem outlined above, and a more general method is needed.

4.3.2 The extraction method

As explained in §4.3.1, direct evaluation of the integrals in (4.1.3) becomes more and more difficult the closer the point of evaluation gets to Γ. The main point about the extraction method is that it completely avoids the use of formula (4.1.3), and does not involve the evaluation of *any* integrals whatsoever. Instead, the key to the method is to use a Taylor series expansion to write the value of $u(\boldsymbol{x})$ for $\boldsymbol{x}$ near Γ in terms of the value of $u(\boldsymbol{x}_0)$, where $\boldsymbol{x}_0$ actually lies *on* Γ, plus some other terms which can be computed.

For example, consider the problem of evaluating u at the point $\boldsymbol{x}$, a distance ϵ from the point $\boldsymbol{x}_0$ on Γ, as shown in Figure 4-1. For simplicity, here we assume that $\boldsymbol{x}-\boldsymbol{x}_0$ is in the normal direction at $\boldsymbol{x}_0$, ie. $\boldsymbol{x}-\boldsymbol{x}_0 = -\epsilon\boldsymbol{n}$, where $\boldsymbol{n}$ is the

unit outward normal vector to Γ.

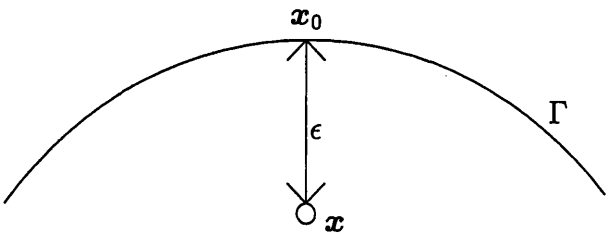

Figure 4-1: Evaluating u at $\boldsymbol{x}$.

Using a Taylor series expansion, we can write $u(\boldsymbol{x})$ in terms of the Dirichlet and Neumann data (one of which will be given and the other of which will have been found, approximately, using the collocation method of Chapter 3), plus some other terms involving the higher order normal derivatives of u at $\boldsymbol{x}_0$,

$$u(\boldsymbol{x}) = u(\boldsymbol{x}_0) - \epsilon \frac{\partial u}{\partial n}(\boldsymbol{x}_0) + \frac{\epsilon^2}{2!}\frac{\partial^2 u}{\partial n^2}(\boldsymbol{x}_0) - \cdots + \frac{(-\epsilon)^r}{r!}\frac{\partial^r u}{\partial n^r}(\boldsymbol{x}_0) + \cdots . \quad (4.3.18)$$

Now as explained in §4.2, in order to guarantee the approximation of u to $\mathcal{O}(h^2)$ accuracy throughout Ω, we need to evaluate u to $\mathcal{O}(h^4)$ accuracy at points near Γ (ie. at the points of $B1 \cup B2$). If $\boldsymbol{x} \in B1 \cup B2$, then clearly we will have $\epsilon = \mathcal{O}(h)$, and hence in order to acheive $\mathcal{O}(h^4)$ accuracy for our approximation to u (4.3.18) we would need to know $\partial^j u(\boldsymbol{x}_0)/\partial n^j$ to $\mathcal{O}(h^{4-j})$ accuracy, for $j = 0, \ldots, 3$. As explained in §4.1, if either the Dirichlet or the Neumann data is given then we can use the collocation method to compute the complementary boundary data to $\mathcal{O}(h^4)$ accuracy without incurring excessive cost. Hence there will be no problem with evaluating whichever of u or $\partial u/\partial n$ is not given to the required accuracy. The question now is how to compute the higher order normal derivatives, and we devote the rest of this section to showing how this can be done.

Note however that (as explained in §4.2), in [59] it is claimed that it is usually sufficient to evaluate u just to $\mathcal{O}(h^2)$ accuracy on $B1 \cup B2$. In this case there would *in principle* be no need to evaluate any of the higher order normal derivatives in (4.3.18). In practice however, as the mesh parameter n increases, ϵ will not decrease uniformly at each point, because the grid R_h does not match up in any way with the boundary Γ, and so even in this case it may still be advisable to

compute some of the higher order normal derivatives of u.

Now we describe how to do this. First, note that differentiating (4.1.1) j times with respect to x, y, it is clear that $\partial^j u/\partial x^j$ and $\partial^j u/\partial y^j$ both satisfy (4.1.1). Using (4.1.2) we then have that

$$\mathcal{L}_\alpha \frac{\partial}{\partial n}\left(\frac{\partial^j u}{\partial x^j}\right) = \frac{1}{2}\frac{\partial^j u}{\partial x^j} + \mathcal{K}_\alpha \frac{\partial^j u}{\partial x^j}, \quad \text{on } \Gamma, \tag{4.3.19}$$

and similar equations for $\partial^j u/\partial y^j$ and $\partial^j u/\partial x^{j-r}\partial y^r$, $r = 1, \ldots, j-1$.

The next thing to note is that we can find $\partial u/\partial \tau$, the tangential derivative of u, to superalgebraic accuracy with respect to n_c (the number of collocation points) using trigonometric interpolation. The cost of doing this at n_c points is $\mathcal{O}(n_c^2 \log n_c)$, and we demonstrate now how this works.

Suppose n_c is even, and define the interpolation points $t_j = 2\pi j/n_c$, $j = -\frac{n_c}{2}+1, \ldots, \frac{n_c}{2}$. It can then be shown (see for example [9, §6.9] or [30, p.35]) that the coefficients c_k of the trigonometric polynomial

$$\phi(s) = \sum_{k=-\frac{n_c}{2}+1}^{\frac{n_c}{2}} c_k e^{iks},$$

which interpolates u at t_j, $j = -\frac{n_c}{2}+1, \ldots, \frac{n_c}{2}$, are given by

$$c_k = \frac{1}{n_c} \sum_{j=-\frac{n_c}{2}+1}^{\frac{n_c}{2}} u(t_j) e^{-ikt_j},$$

and then, moreover,

$$\|u - \phi\|_{H^r} \leq C \frac{1}{n_c^{p-r}} \|u\|_{H^p}, \quad 0 \leq r < p, \quad p > \frac{1}{2}, \tag{4.3.20}$$

where the constant C is independent of n_c and u. Note that this is essentially the same error estimate as (3.3.27), but here we have written the trigonometric interpolating polynomial in a slightly different way. Applying (4.3.20) with $r = 1$

shows that

$$\frac{\partial \phi}{\partial s} = \sum_{k=-\frac{n_c}{2}+1}^{\frac{n_c}{2}} ikc_k e^{iks} \tag{4.3.21}$$

will converge superalgebraically to $\partial u/\partial s = |\gamma'(s)|\partial u/\partial \tau$. The coefficients c_k can each be calculated with $\mathcal{O}(n_c \log n_c)$ cost using the FFT (see for example [9, Chapter 10]). Hence the cost of finding $\partial u/\partial \tau$ at all of the points t_j, $j = -\frac{n_c}{2}+1, \ldots, \frac{n_c}{2}$ is $\mathcal{O}(n_c^2 \log n_c)$. This procedure is sometimes called *Fourier differentiation.*

Now denote the unit outward normal vector to Γ by $\boldsymbol{n} = (n_1, n_2)$, and then the corresponding tangent vector in the anticlockwise direction is $\boldsymbol{\tau} = (-n_2, n_1)$. Using (4.3.19) and (4.3.21) we are then in a position to show how we can find $\partial^2 u/\partial n^2$ given u and $\partial u/\partial n$. We do this using a four step algorithm.

i). Evaluate $\partial u/\partial \tau$ using (4.3.21).

ii). Solve the 2×2 system

$$\begin{pmatrix} n_1 & n_2 \\ -n_2 & n_1 \end{pmatrix} \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial n} \\ \frac{\partial u}{\partial \tau} \end{pmatrix}$$

for $\partial u/\partial x$ and $\partial u/\partial y$.

iii). Solve (4.3.19) with $j = 1$ to find $\frac{\partial}{\partial n}(\partial u/\partial x)$. Solve the analogue of (4.3.19) with x replaced by y to find $\frac{\partial}{\partial n}(\partial u/\partial y)$.

iv). We then can compute the second order normal derivative

$$\frac{\partial^2 u}{\partial n^2} = n_1 \frac{\partial^2 u}{\partial n \partial x} + n_2 \frac{\partial^2 u}{\partial n \partial y},$$

and thus the third term in (4.3.18).

This procedure may be continued to find higher order normal derivatives. For this we need the following result.

Lemma 4.3.3 *For all $j \geq 1$ we have*

$$\frac{\partial^j u}{\partial n^j} = \sum_{r=0}^{j-1} \binom{j-1}{r} n_1^{j-1-r} n_2^r \frac{\partial}{\partial n} \left(\frac{\partial^{j-1} u}{\partial x^{j-1-r} \partial y^r} \right). \tag{4.3.22}$$

Proof: We prove this result by induction. It is clearly true for $j = 1$. Now suppose it is true for some $j \geq 1$. Then

$$\begin{aligned}\frac{\partial^{j+1}u}{\partial n^{j+1}} &= \frac{\partial}{\partial n}\left(\frac{\partial^j u}{\partial n^j}\right) \\ &= n_1^j \frac{\partial^{j+1}u}{\partial n \partial x^j} + \left[\sum_{r=1}^{j-1}\left(\begin{pmatrix} j-1 \\ r \end{pmatrix} + \begin{pmatrix} j-1 \\ r-1 \end{pmatrix}\right) n_1^{j-r} n_2^r \frac{\partial^{j+1}u}{\partial n \partial x^{j-r} \partial y^r}\right] \\ &\quad + n_2^j \frac{\partial^{j+1}u}{\partial n \partial y^j},\end{aligned}$$

under the inductive hypothesis. Now $\begin{pmatrix} j-1 \\ r \end{pmatrix} + \begin{pmatrix} j-1 \\ r-1 \end{pmatrix} = \begin{pmatrix} j \\ r \end{pmatrix}$, and so

$$\frac{\partial^{j+1}u}{\partial n^{j+1}} = \sum_{r=0}^{j}\begin{pmatrix} j \\ r \end{pmatrix} n_1^{j-r} n_2^r \frac{\partial^{j+1}u}{\partial n \partial x^{j-r} \partial y^r},$$

and the result follows. □

We can then find $\partial^j u/\partial n^j$ for all $j \geq 2$ using the formula (4.3.22), where the derivatives on the right hand side are found using similar steps to (i)–(iii) above. So, in principle the Taylor series (4.3.18) for $u(\boldsymbol{x})$ can be computed to arbitrarily high accuracy *without* the computation of integrals such as (4.1.3) for $\boldsymbol{x}$ near Γ. The implementation of this scheme in practice, including a discussion of difficulties such as determining $\boldsymbol{x}_0$ and ϵ for a general boundary, is considered in §4.4.

We now briefly discuss the accuracy and cost of this four step algorithm for finding the higher order normal derivatives. First, note that steps (ii) and (iv) are exact, and step (i) clearly satisfies our cost and accuracy requirements, as it is superalgebraically accurate in terms of n_c, with a cost of only $\mathcal{O}(n_c^2 \log n_c)$. Step (iii) basically consists of solving first kind integral equations of the form (4.1.2) using the collocation method. These solves involve no new technology, as the matrices involved will have already been set up for the original solve of (4.1.2), to find the complementary boundary data. If the LU decompositions of the matrices involved are stored after the original solve, then the cost of further solves will only be $\mathcal{O}(n_c^2)$, and so this part of the algorithm is also very cheap. The overall cost of the four step algorithm will clearly then not be excessive, and using the method

we can find any higher order normal derivatives of u to the same order accuracy as that acheived for the original computation of the complementary boundary data.

Finally, note that as mentioned in §4.1, this method for evaluating u near Γ is similar to that developed by Schulz, Schwab and Wendland in [77] (for Laplace's equation in 2D) and [78] (for Laplace's equation in 3D). There, the same Taylor series representation (4.3.18) is used for u near Γ, but a different method is used for the evaluation of the higher order normal derivatives.

As the cost and accuracy of our method are perfectly acceptable for our needs, we have not made any attempt to directly compare the performance of our method with that of [77].

4.4 Implementation and numerical results

In this section we will first discuss the computational issues which arise in the implementation of the extraction method, and then we will present some numerical results showing how the extraction method outperforms the other schemes described in §4.3.1 for evaluating u near Γ. We will then present further numerical results combining the extraction method with the method of §4.2, illustrating that the accuracy and cost of this method compares favourably with evaluating u throughout Ω by evaluating the integrals (4.1.3) directly at every mesh point.

To implement the extraction method in practice, for a fixed point $\boldsymbol{x}$ near Γ, we first need to do *either* of the following.

i). Find a *nearest* point $\boldsymbol{x}_0$ on the boundary, and the distance ϵ to that point.

ii). Find *any* point $\boldsymbol{y}$ on the boundary such that

$$|\boldsymbol{x}-\boldsymbol{y}| \leq \begin{cases} h, & \text{if } \boldsymbol{x} \in B1, \\ 2h, & \text{if } \boldsymbol{x} \in B2, \end{cases}$$

and then find the value of $|\boldsymbol{x}-\boldsymbol{y}|$, and the direction of $\boldsymbol{y}-\boldsymbol{x}$, so that we can then use the directional derivatives in that direction in place of the normal derivatives in (4.3.18).

The second approach may be easier if the domain Ω is complicated, as we shall show below. However, if it is possible to find the nearest point on the boundary,

then the first approach will give the best convergence, as this minimises the value of ϵ in (4.3.18).

We shall consider the evaluation of $u(\boldsymbol{x})$ at the point $\boldsymbol{x} = (x_1, x_2)$, and we assume that we are given a parametrisation $\boldsymbol{\gamma}(t) = (\gamma_1(t), \gamma_2(t))$, $t \in [0, 2\pi]$, of the boundary Γ.

To find a *nearest* point on the boundary to $\boldsymbol{x}$, note that if $\boldsymbol{y} = (y_1, y_2) = (\gamma_1(t), \gamma_2(t))$ is a nearest point then $\boldsymbol{y} - \boldsymbol{x} = \epsilon \boldsymbol{n}(t)$, where $\boldsymbol{n}(t)$ is the unit outward normal vector to Γ, given by $\boldsymbol{n} = (\gamma_2'(t)/|\boldsymbol{\gamma}'(t)|, -\gamma_1'(t)/|\boldsymbol{\gamma}'(t)|)$. This gives us two simultaneous equations for t and ϵ

$$\begin{aligned} \gamma_1(t) - x_1 &= \epsilon \frac{\gamma_2'(t)}{|\boldsymbol{\gamma}'(t)|} \\ \gamma_2(t) - x_2 &= -\epsilon \frac{\gamma_1'(t)}{|\boldsymbol{\gamma}'(t)|}. \end{aligned}$$

We can eliminate ϵ to get a single nonlinear equation for t

$$\gamma_1'(t)[\gamma_1(t) - x_1] + \gamma_2'(t)[\gamma_2(t) - x_2] = 0, \tag{4.4.23}$$

which can then be solved by some method. If Ω is very simple, eg. a circle, then (4.4.23) can be solved analytically, however if Ω is more complicated then it will be necessary to use an iterative method such as Newton's method to solve (4.4.23). This may increase the cost of the method. Also, for a complicated domain (4.4.23) may have many solutions, and to ensure that a nearest point was chosen all solutions would have to be evaluated, and ϵ determined for each one, and then a nearest point would correspond to the minimum value of ϵ.

The alternative method (ii) is just to find *any* point on the boundary a distance less than or equal to h (or $2h$) from $\boldsymbol{x}$. This could be done either by moving round the boundary in some predefined way until one located a point near enough to $\boldsymbol{x}$, or else by using the fact that $\boldsymbol{x}$ is either an irregular point, or a neighbour of an irregular point, and moving in one of the coordinate directions towards the boundary. This is what we do in practice, and then we re-write (4.3.18) replacing the normal derivatives by the appropriate derivatives in the coordinate directions. These higher order derivatives can be found using the same four step algorithm as that described in §4.3.2 for finding the higher order normal derivatives.

We now present some numerical experiments to demonstrate the cost and

accuracy of the extraction method, and the method of §4.2 for evaluating u quickly throughout Ω.

As a first example, we demonstrate how the extraction method is superior to the quadrature methods of §4.3.1 for evaluating the solution of (4.1.1) at a point x near Γ. We solve exactly the same problem as in examples 4.3.1 and 4.3.2, namely (4.1.1), where Ω is the unit circle with boundary $\gamma(t) = (\cos t, \sin t)$, supplemented by Dirichlet boundary data

$$u|_\Gamma = \frac{1}{2\pi} K_0(\alpha\sqrt{5 - 4\cos t}), \quad t \in [0, 2\pi].$$

The exact solution is then given by (4.3.15). We solve this problem for $\alpha = 1$. First, we compute $\partial u/\partial n$, $\partial^2 u/\partial n^2$ and $\partial^3 u/\partial n^3$ using the four step algorithm described in §4.3.2. The errors between the exact and the approximate values at $t = 0$ are shown in Table 4.3. The superalgebraic convergence rate with respect to n_c can clearly be seen.

n_c	$\partial u/\partial n(t=0)$	$\partial^2 u/\partial n^2(t=0)$	$\partial^3 u/\partial n^3(t=0)$
4	3.885e-3	3.387e-2	1.855e-1
8	1.778e-4	2.983e-3	3.248e-2
16	4.234e-7	1.433e-5	3.258e-4
32	3.604e-12	2.458e-10	1.155e-8
64	1.166e-15	4.260e-14	1.563e-12

Table 4.3: Errors in $\partial u/\partial n$, $\partial^2 u/\partial n^2$ and $\partial^3 u/\partial n^3$ at $t = 0$.

Next, we compute $u(1 - \epsilon, 0)$ using the formula

$$u(1-\epsilon,0) \approx u(0) - \epsilon\frac{\partial u}{\partial n}(0) + \frac{\epsilon^2}{2}\frac{\partial^2 u}{\partial n^2}(0) - \frac{\epsilon^3}{6}\frac{\partial^3 u}{\partial n^3}(0). \tag{4.4.24}$$

The errors between the approximate and exact values of $u(1 - \epsilon, 0)$ for various values of ϵ (as in examples 4.3.1 and 4.3.2) are given in Table 4.4. Comparing these errors to those acheived in Tables 4.1 and 4.2, it is clear that as $\epsilon \to 0$ the extraction method gives much better results. Note however that for fixed ϵ the error cannot decrease beyond a certain limit, corresponding to the missing terms in the Taylor series expansion (4.4.24). For each value of ϵ in Table 4.4 the error for $n_c = 64$ is clearly of order ϵ^4, as we would expect.

n_c	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$	$\epsilon = 0.0001$
4	2.461e-4	3.719e-5	3.868e-6	3.884e-7
8	4.279e-6	1.634e-6	1.763e-7	1.777e-8
16	3.977e-6	3.144e-9	4.163e-10	4.227e-11
32	4.002e-6	4.277e-10	3.962e-14	3.469e-18
64	4.002e-6	4.277e-10	4.309e-14	1.388e-17

Table 4.4: Errors evaluating $u(1-\epsilon, 0)$ using (4.4.24).

As a second example, we combine the extraction method with the method of §4.2 to find the solution throughout Ω of the homogeneous Helmholtz type problem (4.1.1), where now Ω is the ellipse with boundary $\gamma(t) = (3\cos t, \sin t)$, supplemented by Dirichlet data

$$u|_\Gamma = \frac{1}{2\pi} K_0\left(\alpha\sqrt{9\cos^2 t + (\sin t - 2)^2}\right), \quad t \in [0, 2\pi]. \tag{4.4.25}$$

In this case the exact solution is given by

$$u(\boldsymbol{x}) = \frac{1}{2\pi} K_0(\alpha|\boldsymbol{x} - (0,2)|).$$

To solve this problem, we begin by reformulating (4.1.1) as (4.1.2), which is a first kind integral equation for the unknown Neumann data $\partial u/\partial n$. We solve this integral equation using the collocation method of Chapter 3, with n_c collocation points, and N quadrature points on the interval $[0,1]$ (ie. n_c takes the role of n in (3.1.11), and N is as in (3.4.40)–(3.4.41)). The mesh parameters for the evaluation of the integrals in the collocation matrices were chosen as $q_1 = 2$, $q_2 = \alpha$ and $r = 10$ for the evaluation of the double layer integrals (3.4.42), and $q_1 = 4$, $q_2 = \alpha$ and $r = 10$ for the evaluation of the single layer integrals (3.7.61).

Having found an approximation to $\partial u/\partial n$, we then embed Ω in a square of side length 10, and cover the square in the $n \times n$ uniform mesh R_h (2.3.13). We then compute an approximation to $u(\boldsymbol{x})$ for all $\boldsymbol{x} \in B1 \cup B2$ using the extraction method of §4.3.2, where we use the first four terms in the Taylor series expansion (4.3.18), ie. up to and including the $\mathcal{O}(\epsilon^3)$ term.

Using these approximate values of u on $B1 \cup B2$, we then compute an approximation $\tilde{f}_{i,j}$ to $f_{i,j}$, given by (4.2.7), and this gives us the right hand side of the system (4.2.8)–(4.2.9). We then solve this system using the Fast Sine Transform

method to give us an approximation to $u(\boldsymbol{x})$ for all $\boldsymbol{x} \in R_h \cap \Omega$.

We solve this problem for $\alpha = 10$ and $\alpha = 100$, in order to demonstrate the robustness of the method with respect to α. For each example we compute the approximate solution for $n = 2^k - 1$, $k = 5, \ldots, 9$. Initially we choose $n_c = \mathcal{O}(\log n)$ and $N = \mathcal{O}(n^2)$. In this case, as explained in §4.1 (and noting from (3.4.40) and (3.4.41) that the total number of quadrature points m always satisfies $m = \mathcal{O}(N)$), we would expect our approximation to $\partial u/\partial n$ to be $\mathcal{O}(n^{-4})$ accurate. As explained in §4.3.2, we would then expect to get an $\mathcal{O}(n^{-4})$ accurate approximation to u on $B1 \cup B2$, and then as explained in §4.2 we would expect this to give us an $\mathcal{O}(n^{-2})$ accurate approximation to u throughout Ω.

We would then expect the estimated order of convergence (EOC), computed using the formula (2.6.57), to satisfy EOC ≈ -2. The errors and the computed values of EOC at the points $\boldsymbol{x}_1 = (0,0)$, $\boldsymbol{x}_2 = (0, 0.3125)$ and $\boldsymbol{x}_3 = (0, 0.625)$ are shown in Tables 4.5 and 4.6, for $\alpha = 10$ and $\alpha = 100$ respectively.

n	n_c	N	error($\boldsymbol{x}_1$)	EOC	error($\boldsymbol{x}_2$)	EOC	error($\boldsymbol{x}_3$)	EOC
31	14	4	1.136e-9	-2.9	1.267e-8	-2.5	1.249e-7	-2.2
63	17	16	1.521e-10	-1.8	2.260e-9	-1.5	2.738e-8	-0.9
127	20	64	4.306e-11	-2.7	8.269e-10	-3.1	1.489e-8	-4.9
255	23	256	6.413e-12	-2.2	9.587e-11	-2.8	5.152e-10	-0.5
511	25	1024	1.351e-12		1.395e-11		3.762e-10	

Table 4.5: Errors for solving (4.1.1), (4.4.25), with $\alpha = 10$.

n	n_c	N	error($\boldsymbol{x}_1$)	EOC	error($\boldsymbol{x}_2$)	EOC	error($\boldsymbol{x}_3$)	EOC
31	14	4	5.964e-54	-22.3	5.842e-51	-16.3	5.724e-48	-10.3
63	17	16	1.188e-60	-3.1	7.237e-56	-10.2	4.395e-51	-2.2
127	20	64	1.410e-61	-4.0	6.039e-59	-11.8	9.478e-52	-20.7
255	23	256	8.812e-63	-3.7	1.700e-62	-4.8	5.590e-58	-10.0
511	25	1024	6.837e-64		6.077e-64		5.498e-61	

Table 4.6: Errors for solving (4.1.1), (4.4.25), with $\alpha = 100$.

The convergence rate is very erratic, especially for the problem with $\alpha = 100$ for which the exact solution is very close to zero everywhere. However, this erratic convergence is not entirely unexpected, as we are combining a number of quite

separate methods, and if anything the convergence rate seems to be better than $\mathcal{O}(n^{-2})$.

In Table 4.7, we show the operation cost for the various components of the method (applied to the problem with $\alpha = 10$), in terms of the number of floating point operations (flops). We assume that the operation cost is $\mathcal{O}(n^p)$, and then p is computed using the formula (2.6.58). The key to Table 4.7 is as follows.

n	n_c	N	flops1	p	flops2	p	flops3	p
31	14	4	3.897e+6	2.5	4.138e+5	0.9	1.376e+5	2.1
63	17	16	2.193e+7	2.5	7.723e+5	0.8	5.957e+5	2.1
127	20	64	1.201e+8	2.4	1.369e+6	1.0	2.565e+6	2.1
255	23	256	6.321e+8	2.2	2.703e+6	0.9	1.104e+7	2.1
511	25	1024	2.982e+9		5.110e+6		4.749e+7	

Table 4.7: Operation cost for solving (4.1.1), (4.4.25), with $\alpha = 10$.

- flops1 is the cost of setting up the collocation matrices, and finding their LU decompositions. We would expect this to be $\mathcal{O}(Nn_c^2) = \mathcal{O}(n^2 \log^2 n)$, and this can be clearly seen in Table 4.7.

- flops2 is the cost of computing u on $B1 \cup B2$. This consists of finding $\partial u/\partial n$, $\partial^2 u/\partial n^2$ and $\partial^3 u/\partial n^3$, and then evaluating u using the first four terms of (4.3.18) for every mesh point in $B1 \cup B2$. The cost of finding the higher order normal derivatives is only $\mathcal{O}(n_c^2) = \mathcal{O}(\log^2 n)$, once the LU decomposition of the collocation matrices is known, and then as there are $\mathcal{O}(n)$ mesh points in $B1 \cup B2$ we would expect the overall cost to be $\mathcal{O}(n \log^2 n)$. Again, this can be clearly seen in Table 4.7.

- flops3 is the cost of solving the $n \times n$ system (4.2.8)–(4.2.9), using the Fast Sine Transform method, to evaluate the approximation to u throughout $R_h \cap \Omega$. As expected this cost is $\mathcal{O}(n^2 \log n)$.

Comparing the costs, it is clear that the greatest cost is that of setting up the collocation matrices. This cost could be reduced by using a higher order quadrature rule, such as Simpson's rule, on the graded mesh (3.4.40)–(3.4.41) for the evaluation of the integrals in §3.4 and §3.7. However, the $\mathcal{O}(n^2 \log^2 n)$ cost is

only an order of $\log n$ worse than the cost of the rest of the method, and so this does not appear to be a major problem.

Note also that the collocation matrices are dependent only on the shape of the domain, and not on the boundary data. Thus if we wish to solve many homogeneous Helmholtz problems on the same domain, but with different boundary data, as will be the case when we apply this method to the solution of parabolic PDEs, then it will only be necessary to compute the collocation matrices once, and not at each time level. We discuss this further in Chapter 6.

Now in the experiment above, we have chosen n_c and N in such a way that our approximation to u on $B1 \cup B2$ is $\mathcal{O}(n^{-4})$ accurate. However, as mentioned in §4.2, in [59] it is claimed that it is sufficient to evaluate u to only $\mathcal{O}(n^{-3})$ or even $\mathcal{O}(n^{-2})$ accuracy on $B1 \cup B2$ in order to maintain the overall $\mathcal{O}(n^{-2})$ accuracy of the method. To test this theory, we solve the same homogeneous Helmholtz problem again (with $\alpha = 10$), but with a few changes to the method.

Again we choose $n_c = \mathcal{O}(\log n)$, and $N = \mathcal{O}(n^2)$, but now we only use the first three terms in the Taylor series expansion (4.3.18), i.e. up to and including the $\mathcal{O}(\epsilon^2)$ term. We would then expect the approximation to u on $B1 \cup B2$ to be only $\mathcal{O}(n^{-3})$ accurate, in which case, considering (4.2.7), we might expect that our approximation to $f_{i,j}$, which we use on the right hand side of (4.2.8), would only be $\mathcal{O}(n^{-1})$ accurate. The errors and the computed values of EOC at the points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are shown in Table 4.8. The errors are roughly comparable

n	n_c	N	error($\boldsymbol{x}_1$)	EOC	error($\boldsymbol{x}_2$)	EOC	error($\boldsymbol{x}_3$)	EOC
31	14	4	5.368e-10	-4.6	5.366e-9	-4.9	3.563e-8	-0.9
63	17	16	2.172e-11	-1.7	1.845e-10	0.9	1.870e-8	-0.6
127	20	64	6.883e-12	-2.7	3.354e-10	-3.2	1.263e-8	-2.2
255	23	256	1.054e-12	-1.0	3.655e-11	-2.9	2.831e-9	-1.8
511	25	1024	5.435e-13		4.812e-12		7.917e-10	

Table 4.8: Errors for solving (4.1.1), (4.4.25) (with $\alpha = 10$), using only an $\mathcal{O}(n^{-3})$ accurate approximation to u on $B1 \cup B2$.

in magnitude with those acheived in Table 4.5, and in fact are smaller at $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, but the convergence rate is slightly more erratic. However, the convergence rate does appear to be better than $\mathcal{O}(n^{-1})$.

We repeated the experiment, again choosing $n_c = \mathcal{O}(\log n)$, and $N = \mathcal{O}(n^2)$,

but now only using the first two terms in the Taylor series expansion (4.3.18), i.e. up to and including the $\mathcal{O}(\epsilon)$ term. We would then expect the approximation to u on $B1 \cup B2$ to be only $\mathcal{O}(n^{-2})$ accurate, in which case, considering (4.2.7), we might expect that our approximation to $f_{i,j}$, which we use on the right hand side of (4.2.8), would only be $\mathcal{O}(1)$ accurate. The errors and the computed values of EOC at the points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are shown in Table 4.9. The errors are now

n	n_c	N	error($\boldsymbol{x}_1$)	EOC	error($\boldsymbol{x}_2$)	EOC	error($\boldsymbol{x}_3$)	EOC
31	14	4	9.508e-10	-2.6	1.038e-8	-2.2	9.648e-8	-1.8
63	17	16	1.544e-10	-1.2	2.280e-9	-0.7	2.723e-8	0.0
127	20	64	6.799e-11	-1.5	1.390e-9	-1.4	2.768e-8	-1.3
255	23	256	2.369e-11	-1.5	5.170e-10	-1.5	1.094e-8	-1.6
511	25	1024	8.335e-12		1.808e-10		3.569e-9	

Table 4.9: Errors for solving (4.1.1), (4.4.25) (with $\alpha = 10$), using only an $\mathcal{O}(n^{-2})$ accurate approximation to u on $B1 \cup B2$.

greater in magnitude than in both Tables 4.8 and 4.5, and the convergence rate is much more erratic. However, although it is less than $\mathcal{O}(n^{-2})$ (as suggested in [59]) it is better than the $\mathcal{O}(1)$ convergence we may have expected, and in fact appears to be between $\mathcal{O}(n^{-2})$ and $\mathcal{O}(n^{-1})$.

These results do not entirely support the claim in [59] that it is sufficient to approximate u to only $\mathcal{O}(n^{-2})$ accuracy on $B1 \cup B2$ in order to maintain the overall $\mathcal{O}(n^{-2})$ accuracy of the method, but they do indicate that approximating u on $B1 \cup B2$ to $\mathcal{O}(n^{-4})$ accuracy may be more than is required.

However, the extra cost involved in approximating u on $B1 \cup B2$ to $\mathcal{O}(n^{-4})$ accuracy is negligible in the overall context of the method, and so in the numerical examples of Chapter 6 this is what we do.

In our next example, we compare the cost of Mayo's method for extending the solution from $B1 \cup B2$ to the whole of $R_h \cap \Omega$ with the cost of evaluating (4.1.3) directly by the trapezoidal rule at many points in Ω. We use exactly the same problem as in examples 4.3.1 and 4.3.2, ie. (4.1.1) on the unit circle with exact solution given by (4.3.15), and again we embed the unit circle in the square of side length 5 and cover the square in the mesh R_h (2.3.13).

First, we use the exact known solution of u on $B1 \cup B2$ on the right hand side of (4.2.8)–(4.2.9), and we solve this system to extend the solution to the whole

of $R_h \cap \Omega$.

Next, we assume again that the solution is known on $B1 \cup B2$, and we compute an approximation to $u(\boldsymbol{x})$ at all points $\boldsymbol{x} \in R_h \cap \Omega$, $\boldsymbol{x} \notin B1 \cup B2$, by evaluating (4.1.3) directly using the trapezoidal rule with n_c uniformly spaced nodes. We increase n_c until the maximum error is approximately equal to the maximum error acheived by solving (4.2.8)–(4.2.9), although we do not increase n_c beyond $n_c = 512$ because of the computational cost involved. Note that as we are only evaluating $u(\boldsymbol{x})$ for $\boldsymbol{x} \notin B1 \cup B2$, we ensure that all of the points of evaluation lie at least a distance $2h$ from Ω, where h is given by (2.3.12).

The maximum errors in $R_h \cap \Omega$ for each method and the cost of the solves are shown in Table 4.10.

		Direct evaluation of (4.1.3)		Solving (4.2.8)–(4.2.9)	
n	n_c	max. error	flops	max. error	flops
31	16	1.457e-4	4.471e+4	1.144e-4	1.146e+5
63	40	4.617e-5	5.998e+5	3.143e-5	4.948e+5
127	128	1.136e-5	9.112e+6	8.909e-6	2.133e+6
255	256	6.127e-6	7.863e+7	2.353e-6	9.163e+6
511	512	7.662e-6	6.513e+8	6.021e-7	3.921e+7

Table 4.10: Comparing direct evaluation of (4.1.3) with solving (4.2.8)–(4.2.9).

Clearly, once n has increased beyond a certain level, the method of §4.2 is much faster than direct evaluation of (4.1.3) throughout Ω. When n is small, and there are no points of evaluation close to Γ, the direct evaluation of (4.1.3) is very fast, but as n increases, and the mesh points get closer to Γ, we need to increase n_c drastically in order to maintain the accuracy of the method, and this rapidly increases the cost.

As a final numerical example, we combine the methods of Chapters 2, 3 and 4 to solve the inhomogeneous Helmholtz problem

$$-\Delta u(\boldsymbol{x}) + \alpha^2 u(\boldsymbol{x}) = (1+\alpha^2) J_0(|\boldsymbol{x}|), \quad \boldsymbol{x} \in \Omega, \tag{4.4.26}$$

$$u(\boldsymbol{x}) = J_0(|\boldsymbol{x}|), \quad \boldsymbol{x} \in \Gamma, \tag{4.4.27}$$

where Ω is the ellipse with boundary $\gamma(t) = (3\cos t, \sin t)$, $t \in [0, 2\pi]$, and J_0 is the Bessel function of first kind of order zero. The exact solution of this problem

is given by

$$u(\boldsymbol{x}) = J_0(|\boldsymbol{x}|), \quad \boldsymbol{x} \in \Omega.$$

To solve this problem we begin by finding a particular solution u_p of (4.4.26), using the domain embedding method of Chapter 2. Note that (4.4.26) is exactly the same equation as that for which we computed a particular solution in the numerical experiments of §2.6, and so for details of exactly what we did we refer the reader there.

Having computed u_p, we use bilinear interpolation to find the values of u_p on Γ, and then we solve the homogeneous Helmholtz problem

$$-\Delta v(\boldsymbol{x}) + \alpha^2 v(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega, \tag{4.4.28}$$

$$v(\boldsymbol{x}) = J_0(|\boldsymbol{x}|) - u_p(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma. \tag{4.4.29}$$

We do this using exactly the same approach as that described above for solving (4.1.1), (4.4.25). The solution of (4.4.26)–(4.4.27) is then given by $u = u_p + v$.

We solve (4.4.26)–(4.4.27) for $\alpha = 10$ and $\alpha = 100$, in order to demonstrate the robustness of the method with respect to α. For each example we again compute the approximate solution for $n = 2^k - 1$, $k = 5, \ldots, 9$, and we compute the errors and the estimated order of convergence (using formula (2.6.57)) at the points $\boldsymbol{x}_4 = (0,0)$, $\boldsymbol{x}_5 = (1.25,0)$ and $\boldsymbol{x}_6 = (2.5,0)$. The results for $\alpha = 10$ and $\alpha = 100$ are shown in Tables 4.11 and 4.12 respectively.

n	n_c	N	error($\boldsymbol{x}_4$)	EOC	error($\boldsymbol{x}_5$)	EOC	error($\boldsymbol{x}_6$)	EOC
31	14	4	6.057e-5	-1.3	5.544e-4	-2.7	5.578e-4	1.0
63	17	16	2.479e-5	-1.5	8.535e-5	-1.7	1.099e-3	-4.1
127	20	64	8.646e-6	-2.8	2.612e-5	-4.5	6.581e-5	-1.5
255	23	256	1.209e-6	-2.4	1.170e-6	-2.8	2.251e-5	-2.1
511	25	1024	2.317e-7		1.725e-7		5.198e-6	

Table 4.11: Errors for solving (4.4.26)–(4.4.27), with $\alpha = 10$.

As explained in §2.6 we would expect our approximation to u_p to be $\mathcal{O}(n^{-2})$ accurate, and with n_c and N chosen as in Tables 4.11 and 4.12, we would also expect our approximation to v to be $\mathcal{O}(n^{-2})$ accurate. Hence we would expect

n	n_c	N	error($\boldsymbol{x}_4$)	EOC	error($\boldsymbol{x}_5$)	EOC	error($\boldsymbol{x}_6$)	EOC
31	14	4	6.031e-7	-2.0	8.065e-4	-13.1	8.362e-2	1.7
63	17	16	1.525e-7	-2.0	9.459e-8	14.3	2.629e-1	-11.3
127	20	64	3.814e-8	-2.0	1.951e-3	-14.6	1.063e-4	-5.0
255	23	256	9.535e-9	-2.0	7.837e-8	-5.7	3.247e-6	-0.6
511	25	1024	2.384e-9		1.554e-9		2.203e-6	

Table 4.12: Errors for solving (4.4.26)–(4.4.27), with $\alpha = 100$.

that EOC ≈ -2.

The expected order of convergence is broadly acheived, although it is somewhat erratic, especially near Γ when $\alpha = 100$. As we are combining so many separate methods in this experiment, this is not entirely unexpected. Note also that the true solution of (4.4.28)–(4.4.29) will be very highly peaked in a boundary layer when α is large, and so it is inevitable that the errors will be larger there.

Finally, in Table 4.13, we compare the operation costs (in flops) of the various components of the method, when applied to the solution of (4.4.26)–(4.4.27) with $\alpha = 10$. Although the costs of the separate components of the method have already been discussed individually in Chapter 2 and in this chapter, we present them here all in one table so that they can be compared. Assuming that the operation cost of each component of the method is $\mathcal{O}(n^p)$, for some $p > 0$, p is computed using the formula (2.6.58). The key to the table is as follows.

n	n_c	N	flops4	p	flops5	p	flops6	p
31	14	4	3.923e+6	2.5	1.355e+5	2.0	5.519e+5	1.3
63	17	16	2.203e+7	2.5	5.413e+5	2.0	1.371e+6	1.5
127	20	64	1.205e+8	2.4	2.239e+6	2.1	3.949e+6	1.8
255	23	256	6.337e+8	2.2	9.434e+6	2.1	1.381e+7	1.9
511	25	1024	2.989e+9		4.001e+7		5.285e+7	

Table 4.13: Operation cost for solving (4.4.26)–(4.4.27), with $\alpha = 10$.

- flops4 is the cost of computing the irregular points, the distances from the irregular points to Γ, the collocation matrices, and their LU decompositions. In other words this is everything that depends only on the shape of the

domain, and not on the right hand side of (4.4.26) or on the boundary data. Thus all of these operations need only be carried out once if the method is applied to many Helmholtz problems on the same domain with different right hand sides. As expected the cost is $\mathcal{O}(n^2 \log^2 n)$.

- flops5 is the cost of computing the particular solution u_p and its values on Γ. As expected this cost is $\mathcal{O}(n^2 \log n)$.

- flops6 is the cost of finding the solution v of the homogeneous Helmholtz type problem (4.4.28)–(4.4.29) at all points of $R_h \cap \Omega$. As expected this cost is also $\mathcal{O}(n^2 \log n)$.

Clearly the greatest cost is that of setting up the collocation matrices. The costs of finding the particular solution and solving the homogeneous problem are comparable in magnitude, and this is not surprising as both employ a very similar Fast Sine Transform method to do the bulk of the computations.

Chapter 5

Integral equation methods for the heat equation

5.1 Introduction and literature review

In this chapter, we shall consider the use of integral equation methods for solving the initial boundary value problem for the heat equation. We begin, in this section, by demonstrating how the heat equation can be reformulated as a first or second kind integral equation (on the cylindrical boundary in space-time), and presenting a brief literature review outlining some ideas for the numerical solution of these integral equations.

In the remainder of this chapter we then focus on two particular methods in the literature, namely those of Lubich and Schneider [56] (in §5.2) and Chapko and Kress [12] (in §5.3). In particular, we present some new ideas for extending the method of [12], and we show how combining it with some of the ideas in Chapter 3 can lead to improved results. Making use of the link between the methods of [56] and [12] (which we demonstrate in §5.3), we then present an error analysis in §5.4 for our modified version of the method in [12], and in §5.5 we present numerical results demonstrating the improved performance of this method.

In Chapter 6 we will discuss the application of the domain embedding method of Chapters 2–4 to the solution of parabolic PDEs, and another of our goals in this chapter is to outline some of the advantages and disadvantages of that method

when compared with some of those already being used in the literature.

So, we begin by presenting some background on the use of integral equation methods for the solution of the heat equation. The problem we wish to solve is as follows. We seek a function $u \in C(\bar{\Omega} \times [0,T])$ which satisfies

$$\frac{1}{c}\frac{\partial u}{\partial t} = \Delta u, \quad \text{in } \Omega \times (0,T], \tag{5.1.1}$$

with the initial condition

$$u(\cdot,0) = u_0, \quad \text{in } \Omega, \tag{5.1.2}$$

and for simplicity we consider here only the Dirichlet boundary condition

$$u = f, \quad \text{on } \Gamma \times [0,T], \tag{5.1.3}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded and simply connected domain, with boundary $\Gamma \in C^2$. The results given here generalise to Neumann boundary conditions (see below) and mixed boundary conditions, although we avoid this detail here. We assume that $T \in (0,\infty)$, $u_0 \in C(\bar{\Omega})$, $f \in C(\Gamma \times [0,T])$, and that u_0 and f satisfy the compatibility condition $u_0(\cdot) = f(\cdot,0)$ on Γ. We will assume, without loss of generality, that the heat conduction constant $c = 1$, as for any $c > 0$ we can transform (5.1.1) into an equation of the form $\partial u/\partial \tilde{t} = \Delta u$ by making the substitution $\tilde{t} = ct$.

We can solve (5.1.1)–(5.1.3) using boundary integral equation methods, and we now give the basic theory of this method. We begin by defining what is meant by the *fundamental solution* of a parabolic PDE $Lu = 0$. (In (5.1.1) (with $c = 1$) $Lu := \Delta u - \partial u/\partial t$.) This definition (see [19, p.3]) differs slightly from Definition 2.1.1 for the fundamental solution of an elliptic PDE.

Definition 5.1.1 *The* fundamental solution *of a parabolic PDE $Lu = 0$ in $\bar{\Omega} \times [0,T]$ is a function $\Phi(\boldsymbol{x} - \boldsymbol{y}, t - \tau)$ defined for all $\boldsymbol{x}, \boldsymbol{y} \in \bar{\Omega}$, $\tau \in [0,T]$, $t \in (\tau,T]$, which satisfies the following conditions.*

i). For fixed $(\boldsymbol{y},\tau) \in \bar{\Omega} \times [0,T]$ it satisfies, as a function of $(\boldsymbol{x},t) \in \Omega \times (\tau,T]$ the equation $Lu = 0$.

ii). For every function $\psi \in C(\bar{\Omega})$, *if* $\boldsymbol{x} \in \Omega$ *and* $\tau \geq 0$ *then*

$$\lim_{t \to \tau} \int_{\Omega} \Phi(\boldsymbol{x} - \boldsymbol{y}, t - \tau)\psi(\boldsymbol{y})d\boldsymbol{y} = \psi(\boldsymbol{x}).$$

It is then shown in [19, Theorem 8, Chapter 1] that the fundamental solution of the two-dimensional heat equation (5.1.1) (with $c = 1$) is given by

$$k(\boldsymbol{x} - \boldsymbol{y}, t - \tau) := \frac{1}{4\pi(t - \tau)} \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{y}|^2}{4(t - \tau)}\right), \quad t > \tau. \tag{5.1.4}$$

From the definition of the fundamental solution it is clear that for any function $\psi \in C(\bar{\Omega})$, the domain integral

$$\mathcal{P}\psi(\boldsymbol{x}, t) := \int_{\Omega} k(\boldsymbol{x} - \boldsymbol{y}, t)\psi(\boldsymbol{y})d\boldsymbol{y} \tag{5.1.5}$$

is an infinitely differentiable solution of (5.1.1) in $\Omega \times (0, T]$, and it can be continuously extended into $\Omega \times [0, T]$, with initial value

$$\mathcal{P}\psi(\boldsymbol{x}, 0) = \psi(\boldsymbol{x}).$$

Using this, it becomes apparent that we need consider only the special case of the initial boundary value problem with homogeneous initial condition. To see why, note that

$$U(\boldsymbol{x}, t) := \mathcal{P}u_0(\boldsymbol{x}, t) \tag{5.1.6}$$

satisfies both (5.1.1) and (5.1.2). If we can then find a function $u(\boldsymbol{x}, t)$ that satisfies (5.1.1), together with the homogeneous initial condition

$$u(\cdot, 0) = 0, \quad \text{in } \Omega, \tag{5.1.7}$$

and the boundary condition

$$u = \mathcal{P}u_0 - f =: g, \quad \text{on } \Gamma \times [0, T], \tag{5.1.8}$$

then $U - u$ will satisfy (5.1.1)–(5.1.3). Note that for $t > 0$ (5.1.5) has a smooth

integrand.

Now, for any function $\psi \in C(\Gamma \times [0,T])$, we define the *single* and *double layer heat potentials* by

$$\mathcal{V}\psi(\boldsymbol{x},t) := \int_0^t \int_\Gamma k(\boldsymbol{x}-\boldsymbol{y},t-\tau)\psi(\boldsymbol{y},\tau)d\boldsymbol{y}\,d\tau, \quad \boldsymbol{x} \in \Omega,\, t \in (0,T], \tag{5.1.9}$$

and

$$\mathcal{W}\psi(\boldsymbol{x},t) := \int_0^t \int_\Gamma \frac{\partial k(\boldsymbol{x}-\boldsymbol{y},t-\tau)}{\partial n(\boldsymbol{y})}\psi(\boldsymbol{y},\tau)d\boldsymbol{y}\,d\tau, \quad \boldsymbol{x} \in \Omega,\, t \in (0,T], \tag{5.1.10}$$

respectively, where $\partial/\partial n(\boldsymbol{y})$ is the derivative with respect to the unit outward normal at $\boldsymbol{y} \in \Gamma$. From the definition of the fundamental solution, these are both infinitely differentiable solutions of (5.1.1) in $\Omega \times (0,T]$, and they can both be continuously extended into $\Omega \times [0,T]$ with initial values $\mathcal{V}\psi(\boldsymbol{x},0) = \mathcal{W}\psi(\boldsymbol{x},0) = 0$. This is why it is important that we consider the heat equation with homogeneous initial condition, as otherwise we cannot use these integral representations for the solution. We thus have that (5.1.9) and (5.1.10) both solve (5.1.1) and (5.1.7). To get solutions that also satisfy the boundary condition (5.1.8), we need to consider their extensions onto Γ. It is shown in [19, Chapter 5] that the single layer heat potential can be continuously extended from $\Omega \times (0,T]$ into $\bar{\Omega} \times (0,T]$ with limiting value

$$\mathcal{V}\psi(\boldsymbol{x},t) = \int_0^t \int_\Gamma k(\boldsymbol{x}-\boldsymbol{y},t-\tau)\psi(\boldsymbol{y},\tau)d\boldsymbol{y}\,d\tau, \quad \boldsymbol{x} \in \Gamma,\, t \in (0,T], \tag{5.1.11}$$

and it is shown in [49, Theorem 9.5] that the double layer heat potential can be continuously extended from $\Omega \times (0,T]$ into $\bar{\Omega} \times (0,T]$ with limiting value

$$\mathcal{W}\psi(\boldsymbol{x},t) = \int_0^t \int_\Gamma \frac{\partial k(\boldsymbol{x}-\boldsymbol{y},t-\tau)}{\partial n(\boldsymbol{y})}\psi(\boldsymbol{y},\tau)d\boldsymbol{y}\,d\tau - \frac{1}{2}\psi(\boldsymbol{x},t), \quad \boldsymbol{x} \in \Gamma,\, t \in (0,T]. \tag{5.1.12}$$

Using (5.1.12) we have that the double layer heat potential (5.1.10) will satisfy (5.1.1), (5.1.7) and (5.1.8) provided the density $\psi(\boldsymbol{x},t)$ satisfies the second

kind integral equation ([49, Theorem 9.8])

$$\psi(\boldsymbol{x},t) - 2\int_0^t \int_\Gamma \frac{\partial k(\boldsymbol{x}-\boldsymbol{y},t-\tau)}{\partial n(\boldsymbol{y})}\psi(\boldsymbol{y},\tau)d\boldsymbol{y}\,d\tau = -2g(\boldsymbol{x},t), \quad x \in \Gamma,\, t \in (0,T]. \tag{5.1.13}$$

The existence and uniqueness of the solution of the initial boundary value problem for the heat equation follows from the existence and uniqueness of the solution of (5.1.13) (see eg. [49, Theorem 9.9]). A similar theory exists when Ω is an unbounded domain whose complement is bounded and simply connected, and u is required also to satisfy the radiation condition

$$u(\boldsymbol{x},t) \to 0, \quad \text{as } |\boldsymbol{x}| \to \infty, \tag{5.1.14}$$

uniformly with respect to all directions $\boldsymbol{x}/|\boldsymbol{x}|$ and all $t \in [0,T]$.

The single layer heat potential (5.1.9) can also be shown to satisfy (5.1.1), (5.1.7) and (5.1.8), provided the density $\psi(\boldsymbol{x},t)$ satisfies the first kind integral equation

$$\int_0^t \int_\Gamma k(\boldsymbol{x}-\boldsymbol{y},t-\tau)\psi(\boldsymbol{y},\tau)d\boldsymbol{y}\,d\tau = g(\boldsymbol{x},t), \quad x \in \Gamma,\, t \in (0,T]. \tag{5.1.15}$$

A solution theory for this integral equation has been developed much more recently than for (5.1.13), but existence and uniqueness of the solution have been established, see for example [3], [18] or [39].

Similar integral equations can also be formulated when the boundary condition (5.1.3) is replaced by a Neumann boundary condition

$$\frac{\partial u}{\partial n} = f, \quad \text{on } \Gamma \times [0,T], \tag{5.1.16}$$

see [19, Chapter 5] for details. Various direct formulations, where the integral equations only involve the unknown Dirichlet or Neumann data, are also possible, see for example [39]. There it is shown that all of the various boundary integral equation formulations lead to the same unique solution of the original initial boundary value problem for the heat equation.

We will now discuss the numerical solution of the first kind integral equa-

tion (5.1.15), noting that similar methods are applicable to the second kind equation (5.1.13) also. To actually solve (5.1.15), there are several possible approaches. Error estimates for the use of Galerkin methods in both space and time have been derived (see for example [18]), but the computing effort required to set up the matrices can be expensive. The use of collocation methods in both space and time has also been suggested (see for example [37] or [40]), but the analysis is less well developed.

An alternative approach is to couple a boundary integral equation method in space with a different type of discretisation in time. One such method which we will consider in this chapter is that developed by Lubich and Schneider in [56]. There, a convolution quadrature method (developed in [54]) based on a linear multistep method for ordinary differential equations, and requiring knowledge of the Laplace transform of the fundamental solution, is used for the time discretisation. This leads to a triangular system of spatial boundary integral equations, each of which can be solved by reasonably standard methods. We discuss this method further in §5.2.

An error analysis is presented in [56] both for the semidiscrete method, and for the "fully discrete" method based on either Galerkin or collocation approaches for solving the spatial boundary integral equations. Here, fully discrete means that approximation with respect to both space and time is taken into account. However, there is no mention of how to evaluate the boundary integrals which arise using this approach, ie. the analysis is not "fully discrete" in the sense used for integral equations which depend on space alone.

Another idea, known in the literature as Rothe's method, is to approximate the time derivative in (5.1.1) directly by some linear multistep method, leading to a sequence of inhomogeneous elliptic PDEs. A general k-step linear multistep method for the ODE $y'(x) = f(x, y)$ is of the form (see eg. [35, Chapter III])

$$\sum_{j=0}^{k} \gamma_j y_{m+j-k} = h \sum_{j=0}^{k} \beta_j f_{m+j-k}, \quad m = 1, \ldots, n, \tag{5.1.17}$$

where γ_i and β_i are real parameters, $h > 0$ is the step size, n is the number of steps, and $y_i = y(x_i)$, $f_i = f(x_i, y_i)$, $x_i = x_0 + ih$. Applying (5.1.17) to (5.1.1)

(with $c = 1$) we get

$$\sum_{j=0}^{k} \gamma_j u_{m+j-k} = h \sum_{j=0}^{k} \beta_j \Delta u_{m+j-k}, \quad m = 1, \ldots, n, \tag{5.1.18}$$

where now $h > 0$ is the time step and $u_i = u(\boldsymbol{x}, ih)$. Assuming $\beta_k \neq 0$ (ie. the linear multistep method (5.1.17) is implicit), we can rearrange (5.1.18) to get

$$-\Delta u_m + \left(\frac{\gamma_k}{h\beta_k}\right) u_m = \frac{1}{h\beta_k} \sum_{j=0}^{k-1} \left\{ h\beta_j \Delta u_{m+j-k} - \gamma_j u_{m+j-k} \right\}, \quad m = 1, \ldots, n. \tag{5.1.19}$$

Thus we have a series of inhomogeneous Helmholtz equations, where the right hand side of each is given by a function of the solutions on the previous time levels. We can prove that under a certain stability assumption (which we specify below) on the linear multistep method (5.1.17), the operator on the left hand side of (5.1.19) is of exactly the same form as the Helmholtz operator considered in Chapters 2–4, but first we need to make some definitions (see for example [36, Chapter V]).

Definition 5.1.2 *We define the* stability region *of the linear multistep method (5.1.17) to be the set*

$$S := \left\{ \mu \in \mathbb{C} : \begin{array}{c} \textit{all solutions } \xi_j(\mu) \textit{ of } \mu = \frac{\varrho(\xi)}{\sigma(\xi)} \textit{ satisfy } |\xi_j(\mu)| \leq 1 \\ \textit{multiple solutions satisfy } |\xi_j(\mu)| < 1 \end{array} \right\},$$

where

$$\begin{aligned} \varrho(\xi) &:= \gamma_k \xi^k + \gamma_{k-1} \xi^{k-1} + \ldots + \gamma_0 \\ \sigma(\xi) &:= \beta_k \xi^k + \beta_{k-1} \xi^{k-1} + \ldots + \beta_0. \end{aligned}$$

Definition 5.1.3 *A linear multistep method of the form (5.1.17) is defined to be* $A(\alpha)$-stable, $0 < \alpha < \pi/2$, *if*

$$S \supset S_\alpha := \{ \mu : |\arg(-\mu)| < \alpha, \ \mu \neq 0 \},$$

where S is the stability region defined in Definition 5.1.2. A method is defined to be $A(0)$-stable *if it is $A(\alpha)$-stable for some $\alpha > 0$.*

We then have the following result.

Lemma 5.1.4 *Provided the linear multistep method (5.1.17) is $A(0)$-stable, we will always have $\gamma_k/\beta_k > 0$.*

Proof: From Definition 5.1.3 it is clear that a necessary (although not sufficient) condition for $A(0)$-stability is that, for $\mu \in (-\infty, 0)$, all solutions ξ of $\mu = \varrho(\xi)/\sigma(\xi)$ satisfy $|\xi| \leq 1$ (ie. the negative real line lies within the stability region). Now note that

$$\frac{\varrho(\xi)}{\sigma(\xi)} = \frac{\gamma_k + \gamma_{k-1}\xi^{-1} + \ldots + \gamma_0\xi^{-k}}{\beta_k + \beta_{k-1}\xi^{-1} + \ldots + \beta_0\xi^{-k}} \to \frac{\gamma_k}{\beta_k}, \text{ as } \xi \to \infty,$$

and so if $\gamma_k/\beta_k < 0$ the method cannot be $A(0)$-stable. Thus $\gamma_k/\beta_k \geq 0$, and under the assumption that $\gamma_k \neq 0$ the result follows. □

So, if we discretise (5.1.1) by an implicit $A(0)$-stable linear multistep method, then using Lemma 5.1.4 we see that the left hand side of (5.1.19) is of the form

$$-\Delta u_m + \alpha^2 u_m, \quad m = 1, \ldots, n,$$

where $\alpha = \sqrt{\gamma_k/h\beta_k}$ is real. The question now is how to solve the series of inhomogeneous Helmholtz equations (5.1.19). One possible approach is to use the domain embedding method of Chapters 2–4 to solve each equation (5.1.19) in turn, and we discuss this method further in Chapter 6. As we shall see there, this approach allows us to

i). deal with the inhomogeneities on the right hand side of (5.1.19) without needing to use domain integrals, and thus it also extends to inhomogeneous and even nonlinear versions of (5.1.1);

ii). take a variable time step h;

iii). solve the initial boundary value problem (5.1.1)–(5.1.3) with an inhomogeneous initial condition ((5.1.2) with $u_0 \neq 0$), without the need to incorporate a domain integral of the form (5.1.6).

Here we discuss some alternative approaches in the literature, and we consider to what extent they satisfy conditions (i)–(iii).

In [46], Kesavan and Vasudevamurthy prove convergence results for the time discretisation scheme (5.1.19), where (5.1.17) is either the backward Euler or Crank-Nicolson method. However, they only consider the one-dimensional case $\Omega \subset \mathbb{R}$. In [21], Gerdes considers the use of Rothe's method for solving the three-dimensional heat equation. However, to solve the inhomogeneous elliptic PDEs he reduces them to homogeneous problems by the use of domain integrals, an approach which loses many of the advantages of the boundary integral equation method, as discussed in §1.1.

In [74], Ramesh and Lean consider the use of the backward Euler method for the time discretisation of the two-dimensional heat equation. Again, the inhomogeneities on the right hand side of (5.1.19) are removed by the use of domain integrals. Ramesh and Lean's scheme for solving the resulting homogeneous problems was discussed in §3.1.

Another paper using the backward Euler method for the time discretisation of the two-dimensional heat equation is that of Chapko and Kress [12]. They avoid the use of domain integrals by finding a fundamental solution to the whole sequence (5.1.19), and the problem then reduces to solving a sequence of first or second kind integral equations of the type described in Chapter 3. This method does however require a homogeneous initial condition, and in [12] only the case of a fixed time step is considered. We discuss the method further in §5.3, and we also propose a simple modification to the method which allows the use of a variable time step.

The method of [12] has also been applied to the solution of the heat equation in [38]. Whereas in [12] only domains with smooth boundaries are considered, in [38] the method is extended to the solution of problems on domains with nonsmooth boundaries.

Although Lubich and Schneider [56] and Chapko and Kress [12] approach the time discretisation of (5.1.1) in different ways, it is shown in [56, Theorem 4.3] that the two semidiscrete methods do in fact lead to identical solutions. We discuss this link between the two schemes further in §5.3.

To implement a fully discrete method, it is necessary to consider methods for solving the spatial boundary integral equations. The suggested method in [12] is

based on a splitting of the kernels similar to (3.1.16) to remove the logarithmic singularities. As described in §3.1, this approach may fail completely when the time step becomes small. To get around this, we propose the use of the collocation method of Chapter 3 for solving the integral equations.

Making use of the link between the methods of [56] and [12], we discuss in §5.4 the error analysis of this combined method, and then in §5.5 we present some numerical results demonstrating how when the time step becomes small the fully discrete method as described in [12] blows up, whilst the method combining our new collocation method of Chapter 3 with the approach of [12] continues to perform well as the time step approaches zero.

Finally, note that in the literature Rothe's method has also been applied to the solution of some nonlinear parabolic PDEs, see for example Kačur and Van Keer [45], Lubich and Ostermann [55]. However, in [45] only the use of finite element methods for the solution of the resulting spatial problems has been considered, and in [55] the spatial discretisation of the problem is not considered at all.

5.2 Time discretisation of heat potentials

In this section, we describe the method of [56] for the time discretisation of the first kind equation (5.1.15), which is of Volterra type with respect to t. The method basically involves the convolution quadrature approximation of the single layer heat potential (5.1.9), and could equally well be applied to the double layer heat potential (5.1.10), and hence to the solution of the second kind equation (5.1.13). First, in §5.2.1 we outline the convolution quadrature method of [54], and then in §5.2.2 we show how it can be applied to the solution of (5.1.15).

5.2.1 Convolution quadrature

The convolution quadrature method which we outline here is for the numerical approximation of convolution integrals of the form

$$(f * g)(x) = \int_0^x f(x-s)g(s)ds, \quad x \geq 0. \tag{5.2.20}$$

We derive below a method for the numerical discretisation of these integrals which is based on the application of the Laplace transform to (5.2.20), approximation in transform space, and then inverse transform. The result is the "convolution quadrature" method of Lubich [54], and the end result is the approximation of (5.2.20)

$$(f * g)_h(x) = \sum_{0 \le jh \le x} \omega_j(h) g(x - jh), \tag{5.2.21}$$

where $h > 0$ is the stepsize and the weights $\omega_j(h)$ are related to the Laplace transform F of f in a way to be made precise below (5.2.30).

To derive (5.2.21) let us consider (5.2.20) and assume that $g \in C[0, x]$ and $f(t)$ is analytic and of (at most) exponential growth as $t \to \infty$, and satisfies $f(t) = \mathcal{O}(t^{\mu-1})$ as $t \to 0$, where $\mu > 0$. We then have that the Laplace transform

$$F(\lambda) = \int_0^\infty f(t) e^{-\lambda t} dt$$

of $f(t)$ is analytic in a sector $|\arg(\lambda - c)| < \pi - \varphi$, with $0 < \varphi < \pi/2$, $c \in \mathbb{R}$, and satisfies there

$$|F(\lambda)| \le M |\lambda|^{-\mu}, \tag{5.2.22}$$

for some $M < \infty$ (see for example [65, Theorem 3.3, p.114]). We then also have the Laplace inversion formula

$$f(t) = \frac{1}{2\pi i} \int_\Sigma F(\lambda) e^{\lambda t} d\lambda, \quad (t > 0), \tag{5.2.23}$$

where Σ runs from $\infty . e^{-i(\pi-\varphi)}$ to $\infty . e^{i(\pi-\varphi)}$ within the sector of analyticity of $F(\lambda)$ (see for example [65, Theorem 11.1, p.315]).

To approximate (5.2.20), we make the substitution $t = x - s$, and then using

the formula (5.2.23) we can write (5.2.20) as

$$\begin{aligned}\int_0^x f(x-s)g(s)ds &= \int_0^x f(t)g(x-t)dt \\ &= \int_0^x \left\{\frac{1}{2\pi i}\int_\Sigma F(\lambda)e^{\lambda t}d\lambda\right\} g(x-t)dt \\ &= \frac{1}{2\pi i}\int_\Sigma F(\lambda)\int_0^x e^{\lambda t}g(x-t)dt\,d\lambda. \qquad (5.2.24)\end{aligned}$$

Now, consider the ordinary differential equation

$$y'(x) = \lambda y(x) + g(x), \quad y(0) = 0. \qquad (5.2.25)$$

Using an integrating factor and a change of variables, it is clear that this has the solution

$$y(x) = \int_0^x e^{\lambda t}g(x-t)dt, \qquad (5.2.26)$$

which is the inner integral in (5.2.24). Hence to approximate (5.2.26) we can proceed indirectly by finding a discretisation of (5.2.25). To do this we use an implicit $A(0)$-stable linear multistep method (see Definition 5.1.3) of the form

$$\sum_{j=0}^{k}\gamma_j y_{m+j-k} = h\sum_{j=0}^{k}\beta_j(\lambda y_{m+j-k} + g_{m+j-k}), \quad m \geq 0, \qquad (5.2.27)$$

with starting values $y_{-k} = \ldots = y_{-1} = 0$, and with g extended by zero to the negative real axis, where $h > 0$ is the time-step, $y_n = y(nh)$, and $g_n = g(nh)$. Multiplying (5.2.27) by ξ^m and summing from $m = 0$ to ∞ we get

$$\sum_{j=0}^{k}\gamma_j\xi^{k-j}\sum_{m=0}^{\infty}y_{m+j-k}\xi^{m+j-k} = h\sum_{j=0}^{k}\beta_j\xi^{k-j}\sum_{m=0}^{\infty}(\lambda y_{m+j-k} + g_{m+j-k})\xi^{m+j-k},$$

and hence

$$\sum_{j=0}^{k}\gamma_j\xi^{k-j}\sum_{n=j-k}^{\infty}y_n\xi^n = h\sum_{j=0}^{k}\beta_j\xi^{k-j}\sum_{n=j-k}^{\infty}(\lambda y_n + g_n)\xi^n.$$

Using the fact that $y_{-k} = \ldots = y_{-1} = 0$ and $g_{-k} = \ldots = g_{-1} = 0$, we then have

$$\sum_{j=0}^{k} \gamma_j \xi^{k-j} \sum_{n=0}^{\infty} y_n \xi^n = h \sum_{j=0}^{k} \beta_j \xi^{k-j} \sum_{n=0}^{\infty} (\lambda y_n + g_n) \xi^n,$$

and so

$$(\gamma_0 \xi^k + \ldots + \gamma_k) \sum_{n=0}^{\infty} y_n \xi^n = (\beta_0 \xi^k + \ldots + \beta_k) \left(h\lambda \sum_{n=0}^{\infty} y_n \xi^n + h \sum_{n=0}^{\infty} g_n \xi^n \right).$$

Solving for $\sum_{n=0}^{\infty} y_n \xi^n$ we get

$$\sum_{n=0}^{\infty} y_n \xi^n = \sum_{n=0}^{\infty} g_n \xi^n \left[\frac{1}{h} \left(\frac{\gamma_0 \xi^k + \ldots + \gamma_k}{\beta_0 \xi^k + \ldots + \beta_k} \right) - \lambda \right]^{-1},$$

and so y_n, the approximate solution of (5.2.25) at the n^{th} time level, is the coefficient of ξ^n in the expansion in increasing powers of ξ of the function

$$\left(\frac{\delta(\xi)}{h} - \lambda \right)^{-1} \sum_{n=0}^{\infty} g_n \xi^n,$$

where $\delta(\xi) = (\gamma_0 \xi^k + \ldots + \gamma_k)/(\beta_0 \xi^k + \ldots + \beta_k)$. Substituting this into (5.2.24), we can define an approximation to (5.2.20) at $x = nh$ to be the coefficient of ξ^n in the expansion of

$$\frac{1}{2\pi i} \int_{\Sigma} F(\lambda) \left(\frac{\delta(\xi)}{h} - \lambda \right)^{-1} \sum_{n=0}^{\infty} g_n \xi^n d\lambda = \sum_{n=0}^{\infty} g_n \xi^n F\left(\frac{\delta(\xi)}{h} \right), \tag{5.2.28}$$

where the equality arises from Cauchy's integral formula (it is shown that this is valid in [54]). Defining $\omega_j(h)$ to be the coefficients of the power series

$$F\left(\frac{\delta(\xi)}{h} \right) = \sum_{j=0}^{\infty} \omega_j(h) \xi^j, \tag{5.2.29}$$

ie.

$$\omega_j(h) = \frac{1}{j!} \frac{\partial^j}{\partial \xi^j} F\left(\frac{\delta(\xi)}{h} \right) \Big|_{\xi=0}, \quad j = 0, 1, 2, \ldots, \tag{5.2.30}$$

and substituting (5.2.29) into (5.2.28), we get

$$\sum_{n=0}^{\infty} g_n \xi^n F\left(\frac{\delta(\xi)}{h}\right) = \sum_{n=0}^{\infty} g_n \xi^n \sum_{j=0}^{\infty} \omega_j(h) \xi^j,$$

and replacing $n+j$ by n in this we see that the coefficient of ξ^n in (5.2.28) is then given by

$$\sum_{j=0}^{n} \omega_j(h) g_{n-j} = \sum_{0 \le jh \le x} \omega_j(h) g(x - jh)$$

which is the quantity $(f * g)_h$ defined in (5.2.21).

Under appropriate assumptions on the linear multistep method (5.2.27) (see [54] for details) we then have the following error estimate for the convolution quadrature approximation (5.2.21). It appears in [54] as Theorem 3.1, and we quote it here without proof.

Theorem 5.2.1 *For $g \in C^p[0, \bar{x}]$, ($\bar{x} < \infty$ fixed), and for all $x \in [h, \bar{x}]$, there exists a constant C independent of h, x and g such that*

$$\begin{aligned} |(f * g)_h(x) - (f * g)(x)| \;\le\; & C x^{\mu-1} \{h|g(0)| + \ldots + h^{p-1}|g^{(p-2)}(0)| \\ & + h^p(|g^{(p-1)}(0)| + x \max_{0 \le t \le x} |g^{(p)}(t)|)\}, \end{aligned} \tag{5.2.31}$$

where p is the order of the linear multistep method (5.2.27), and μ is given in (5.2.22).

Remark: From (5.2.31) it would appear that the convolution quadrature approximation (5.2.21) converges with only $\mathcal{O}(h)$ accuracy. However, when the order of the linear multistep method $p \ge 2$, the full order of convergence of the linear multistep method (ie. $\mathcal{O}(h^p)$) can be acheived by adding to $(f * g)_h(x)$ a suitable linear combination of g_j, $j = 0, \ldots, p-1$, see [54] for details.

In practice, we will only be considering this method with respect to its relationship with the method in [12], and so we need only consider the first order backward Euler method, which is given by (5.2.27) with $k = 1$, $\gamma_0 = -1$, $\gamma_1 = 1$, $\beta_0 = 0$ and $\beta_1 = 1$, and for which no modifications are needed.

5.2.2 Discretising (5.1.15) in time using the convolution quadrature method

In this section we show how (5.1.15) can be discretised in time, as described in [56]. The left hand side of (5.1.15) is just the single layer heat potential (5.1.9), which we approximate using the convolution quadrature method of §5.2.1. Note that this method could also be used for the approximation of (5.1.10), and thus for the solution of (5.1.13), but for simplicity we just present the method as applied to (5.1.15). The time-dependent part of the left hand side of (5.1.15) is of the form (5.2.20), with $k(\boldsymbol{x}-\boldsymbol{y},\cdot)$ and $\psi(\boldsymbol{y},\cdot)$ taking the roles of f and g respectively. With $\psi \in \mathcal{C}(\Gamma \times [0,T])$, all of the assumptions on f and g in (5.2.20) are clearly satisfied. To apply the rule (5.2.21) we then just need to compute the Laplace transform (with respect to t) of the kernel $k(\boldsymbol{x},t)$. This is done in the following lemma.

Lemma 5.2.2 *The Laplace transform of $k(\boldsymbol{x},t)$ for $\lambda > 0$ is the fundamental solution of the Helmholtz equation $\Delta u - \lambda u = 0$, which is given by*

$$K(\boldsymbol{x},\lambda) = \frac{1}{2\pi} K_0(\sqrt{\lambda}\,|\boldsymbol{x}|). \tag{5.2.32}$$

Proof: First note that as $k(\boldsymbol{x},t)$ is the fundamental solution of the heat equation (5.1.1), we have from Definition 5.1.1 that

$$\left(\frac{\partial}{\partial t} - \Delta\right) k(\boldsymbol{x},t) = 0, \quad \boldsymbol{x} \in \Omega,\, t > 0, \tag{5.2.33}$$

and

$$\lim_{t\to 0} \int_\Omega k(\boldsymbol{x}-\boldsymbol{y},t)\varphi(\boldsymbol{y})d\boldsymbol{y} = \varphi(\boldsymbol{x}), \quad \varphi \in \mathcal{C}(\bar{\Omega}),\, \boldsymbol{x} \in \Omega,$$

which implies that $k(\boldsymbol{x}-\boldsymbol{y},0) = \delta(\boldsymbol{x}-\boldsymbol{y})$, where δ represents the dirac delta function. We now define the Laplace transform of $k(\boldsymbol{x},t)$ by

$$K(\boldsymbol{x},\lambda) = \mathcal{L}\{k(\boldsymbol{x},\cdot)\}(\lambda),$$

and then we take the Laplace transform of $\partial k/\partial t$ and integrate by parts to get

$$\begin{aligned}\mathcal{L}\left\{\frac{\partial k}{\partial t}(\boldsymbol{x},\cdot)\right\}(\lambda) &= \int_0^\infty e^{-\lambda t}\frac{\partial k}{\partial t}(\boldsymbol{x},t)dt \\ &= \left[e^{-\lambda t}k(\boldsymbol{x},t)\right]_0^\infty + \lambda\int_0^\infty e^{-\lambda t}k(\boldsymbol{x},t)dt \\ &= k(\boldsymbol{x},0) + \lambda K(\boldsymbol{x},\lambda).\end{aligned}$$

Now using (5.2.33), and noting that $\mathcal{L}\{\Delta k(\boldsymbol{x},\cdot)\}(\lambda) = \Delta K(\boldsymbol{x},\lambda)$ (where here Δ denotes the Laplacian with respect to $\boldsymbol{x}$), we get

$$\Delta K(\boldsymbol{x},\lambda) - \lambda K(\boldsymbol{x},\lambda) = k(\boldsymbol{x},0) = \delta(\boldsymbol{x}).$$

Recalling Definition 2.1.1 the result follows. □

Remark: Using the ascending series expansion for $K_0(z)$ (see Appendix B), it can be easily shown that $K(\boldsymbol{x},\lambda)$ satisfies (5.2.22) for all $\mu > 0$.

Applying the convolution quadrature approximation (5.2.21) to (5.1.9) (for $t_n = (n+1)h$, $n = 0,\ldots,N-1$), and using Lemma 5.2.2, we then get

$$\mathcal{V}\psi(\boldsymbol{x},t_n) \approx \int_\Gamma \left[\sum_{j=0}^{n}\omega_{n-j}(h)\psi(\boldsymbol{y},jh)\right]d\boldsymbol{y}, \quad n = 0,\ldots,N-1, \tag{5.2.34}$$

where N is the number of time steps $h = T/N$, and $\omega_j(h)$, $j = 0,\ldots,N-1$, are the coefficients of the power series

$$K\left(\boldsymbol{x}-\boldsymbol{y},\frac{\delta(\xi)}{h}\right) = \sum_{j=0}^{\infty}\omega_j(h)\xi^j.$$

Thus, recalling (5.2.30),

$$\omega_j(h) = \frac{1}{j!}\frac{\partial^j}{\partial\xi^j}K\left(\boldsymbol{x}-\boldsymbol{y},\frac{\delta(\xi)}{h}\right)\Big|_{\xi=0}, \quad j = 0,1,2,\ldots,N-1,$$

and hence we can write (5.2.34) as

$$\mathcal{V}\psi(\boldsymbol{x},t_n) \approx \sum_{j=0}^{n}\omega_{n-j}(h,V)\psi(\boldsymbol{x},jh), \quad n = 0,\ldots,N-1, \tag{5.2.35}$$

where now $\omega_j(h,V)$, $j=0,\ldots,N-1$, are operators given by

$$\omega_j(h,V)\phi(\boldsymbol{x}) = \frac{1}{2\pi}\int_\Gamma \tilde{\Psi}_j(\boldsymbol{x},\boldsymbol{y})\phi(\boldsymbol{y})d\boldsymbol{y}, \quad \boldsymbol{x}\in\Gamma, \tag{5.2.36}$$

with (recalling (5.2.32))

$$\tilde{\Psi}_j(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{j!}\frac{\partial^j}{\partial\xi^j}K_0\left(\sqrt{\frac{\delta(\xi)}{h}}|\boldsymbol{x}-\boldsymbol{y}|\right)\Bigg|_{\xi=0}. \tag{5.2.37}$$

For $\psi\in\mathcal{C}^p(\Gamma\times[0,T])$, where p is the order of the linear multistep method (5.2.27), we can then use Theorem 5.2.1 and the remark after Lemma 5.2.2 to get the error bound

$$\left|\mathcal{V}\psi(\boldsymbol{x},t_n) - \sum_{j=0}^{n}\omega_{n-j}(h,V)\psi(\boldsymbol{x},t_j)\right| \leq$$
$$C\left\{h|\psi(\boldsymbol{x},0)|+\ldots+h^{p-1}|\psi^{p-2}(\boldsymbol{x},0)|+h^p\left(|\psi^{p-1}(\boldsymbol{x},0)|+t_n\max_{0\leq\tau\leq t_n}|\psi^p(\boldsymbol{x},\tau)|\right)\right\},$$

for all $n=0,\ldots,N-1$, where the constant C is independent of $\boldsymbol{x}$.

We then solve (5.1.15) by substituting (5.2.35) into the left hand side, for $t_n=(n+1)h$, $n=0,\ldots,N-1$, to get the series of equations

$$\sum_{j=0}^{n}\omega_{n-j}(h,V)\psi_j(\boldsymbol{x}) = g(\boldsymbol{x},t_n), \quad n=0,\ldots,N-1. \tag{5.2.38}$$

To find $\psi_n(\boldsymbol{x})=\psi(\boldsymbol{x},t_n)$, $n=0,\ldots,N-1$, we then only need to invert $\omega_0 = V(\delta(0)/h)$ on each level, ie. we need to solve

$$\omega_0(h,V)\phi_n(\boldsymbol{x}) = g(\boldsymbol{x},t_n) - \sum_{j=0}^{n-1}\omega_{n-j}(h,V)\phi_j(\boldsymbol{x}), \tag{5.2.39}$$

for $n=0,\ldots,N-1$. Once $\psi_n(\boldsymbol{x})$, $n=0,\ldots,N-1$, are found, the semidiscrete (ie. with respect to time only) approximation to the solution of the heat equation

is then given by

$$u_N(\boldsymbol{x}, t_n) = \sum_{j=0}^{n} \omega_{n-j}(h, U)\phi_j(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{5.2.40}$$

where $\omega_j(h, U)$ are given by exactly the same formula (5.2.36) as $\omega_j(h, V)$, but now with $\boldsymbol{x} \in \Omega$.

So, we have shown how we can find an approximation $u_N(\boldsymbol{x}, t_n)$ (5.2.40) to the true solution $u(\boldsymbol{x})$ of (5.1.1), (5.1.7) and (5.1.8) by using the convolution quadrature approach to approximate (5.1.15) directly (the *convolution quadrature method*). However, it is in fact shown in [56] that this method leads to exactly the same solution as if we applied the linear multistep method (5.2.27) to (5.1.1) directly, as in (5.1.19) (*Rothe's method*). This result and also some error bounds for these two equivalent semidiscrete approximations of u are summarised in the following theorem, which appears as [56, Theorem 4.3], and which we quote here without proof.

Theorem 5.2.3 *If g in (5.1.15) satisfies $g(\cdot, t) \in H^{1/2}(\Gamma)$ for all $t \geq 0$ and $\|g(\cdot, t)\|_{H^{1/2}(\Gamma)} \in C^2(0, T)$, then the convolution quadrature approximation u_N given by (5.2.40) is* identical *to the semidiscrete solution obtained by Rothe's method of applying the linear multistep method (5.2.27) directly to (5.1.1) as in (5.1.19) (assuming the spatial equations are solved exactly). Also, at $t_n = (n+1)h$, $n = 0, \ldots, N-1$, we have the error bounds*

$$\begin{aligned} \|u_N(\cdot, t_n) - u(\cdot, t_n)\|_{H^1(\Omega)} \leq\ & Cht_n^{-\frac{3}{2}} \left(\|g(\cdot, 0)\|_{H^{1/2}(\Gamma)} + t_n \left\| \left. \frac{\partial g(\cdot, t)}{\partial t} \right|_{t=0} \right\|_{H^{1/2}(\Gamma)} \right. \\ & \left. + \frac{t_n^2}{2} \max_{0 \leq s \leq t_n} \left\| \left. \frac{\partial^2 g(\cdot, t)}{\partial t^2} \right|_{t=s} \right\|_{H^{1/2}(\Gamma)} \right) \end{aligned} \tag{5.2.41}$$

$$\begin{aligned} \|u_N(\cdot, t_n) - u(\cdot, t_n)\|_{H^{-1}(\Omega)} \leq\ & Cht_n^{-\frac{1}{2}} \left(\|g(\cdot, 0)\|_{H^{1/2}(\Gamma)} + t_n \left\| \left. \frac{\partial g(\cdot, t)}{\partial t} \right|_{t=0} \right\|_{H^{1/2}(\Gamma)} \right. \\ & \left. + \frac{t_n^2}{2} \max_{0 \leq s \leq t_n} \left\| \left. \frac{\partial^2 g(\cdot, t)}{\partial t^2} \right|_{t=s} \right\|_{H^{1/2}(\Gamma)} \right). \end{aligned} \tag{5.2.42}$$

So, although Rothe's method and the convolution quadrature method appear to approach the problem from different directions, they do in fact lead to identical solutions (assuming the spatial equations are solved exactly). This means that if we approximate (5.1.1) directly using a linear multistep method, leading to the series of equations (5.1.19) (supplemented by the appropriate initial and boundary conditions), then we can use the error bounds of Theorem 5.2.3 to bound the temporal error in our approximation, regardless of how we actually solve the spatial equations.

In the next section we shall discuss an approach for solving (5.1.1), (5.1.7) and (5.1.8) based on the direct time discretisation of (5.1.1) using the backward Euler method. In §5.4 we then discuss how we can combine Theorem 5.2.3 with a bound on the errors incurred in solving the resulting spatial equations (5.1.19) in order to get a fully discrete error bound for this method.

5.3 Implementing Rothe's method for the heat equation using boundary integrals

In this section we obtain a sequence of boundary integral equations in space for the initial boundary value problem (5.1.1), (5.1.7) and (5.1.8) (and (5.1.14) if Ω is unbounded), after discretisation in time with the backward Euler method. This method was also discussed in [12] where only a fixed time step was considered for the time discretisation. Here, we will also show how the method can be adapted to allow for a variable time step.

We also describe in this section how to discretise the resulting boundary integral equations in space. The method suggested in [12] for this purpose is significantly flawed, with the errors blowing up as the time step approaches zero, as we shall demonstrate in §5.5. We use instead the method described in Chapter 3.

Discretising (5.1.1) in time using the backward Euler method with a variable time step leads to the sequence of inhomogeneous Helmholtz equations

$$-\Delta u_n + \alpha_n^2 u_n = \alpha_n^2 u_{n-1}, \quad \text{in } \Omega, \quad n = 0, \ldots, N-1, \tag{5.3.43}$$

where $\alpha_n = 1/\sqrt{h_n}$, N is the number of time steps $h_n > 0$ satisfying $\sum_{n=0}^{N-1} h_n = T$, $t_n = \sum_{j=0}^{n} h_j$, and $u_n = u(\cdot, t_n)$. From (5.1.7) we also have the initial condition

$$u_{-1} = 0, \quad \text{in } \Omega, \tag{5.3.44}$$

and from (5.1.8) we have the boundary condition

$$u_n = g_n := g(\cdot, t_n), \quad \text{on } \Gamma, \quad n = 0, \ldots, N-1. \tag{5.3.45}$$

Now (5.3.43)–(5.3.45) can be thought of as the boundary value problem

$$A\tilde{u} = 0, \quad \text{in } \Omega, \tag{5.3.46}$$

$$\tilde{u} = \tilde{g}, \quad \text{on } \Gamma, \tag{5.3.47}$$

for the unknown sequence $\tilde{u} = (u_0, \ldots, u_{N-1})^T$, where $\tilde{g} = (g_0, \ldots, g_{N-1})^T$ and

$$A = \begin{pmatrix} -\Delta + \alpha_0^2 & 0 & 0 & \cdot & 0 \\ -\alpha_1^2 & -\Delta + \alpha_1^2 & 0 & \cdot & 0 \\ 0 & -\alpha_2^2 & -\Delta + \alpha_2^2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & -\alpha_{N-1}^2 & -\Delta + \alpha_{N-1}^2 \end{pmatrix}.$$

Solving this system is done in two stages. First, a fundamental solution

$$\Psi(\boldsymbol{x}, \boldsymbol{y}) = (\Psi_0(\boldsymbol{x}, \boldsymbol{y}), \ldots, \Psi_{N-1}(\boldsymbol{x}, \boldsymbol{y}))^T \tag{5.3.48}$$

of (5.3.46) is constructed, with $\Psi(\boldsymbol{x}, \boldsymbol{y})$ satisfying (5.3.46) for all $\boldsymbol{x} \neq \boldsymbol{y}$, and blowing up logarithmically as $\boldsymbol{x} \to \boldsymbol{y}$. The solution of (5.3.46) and (5.3.47) is then sought as a single or double layer potential, whose densities can be found by solving a sequence of first or second kind integral equations which arise from (5.3.47). We consider these two stages separately.

5.3.1 Constructing a fundamental solution

Recalling formula (2.1.5) for the fundamental solution of the Helmholtz equation, it is clear that $\Psi_0(\boldsymbol{x}, \boldsymbol{y}) = cK_0(\alpha_0|\boldsymbol{x} - \boldsymbol{y}|)$ (where $c \neq 0$ is any constant) satis-

fies (5.3.43) for $n = 0$, $\boldsymbol{x} \neq \boldsymbol{y}$, and it also blows up logarithmically as $\boldsymbol{x} \to \boldsymbol{y}$ (see Appendix B). Recalling also the formulae for the derivatives of the modified Bessel functions of order zero and one (B.0.4), we seek Ψ_n, $n = 1, \ldots, N-1$, in the form

$$\Psi_n(\boldsymbol{x}, \boldsymbol{y}) = K_0(\alpha_0|\boldsymbol{x} - \boldsymbol{y}|)v_n(|\boldsymbol{x} - \boldsymbol{y}|) + K_1(\alpha_0|\boldsymbol{x} - \boldsymbol{y}|)w_n(|\boldsymbol{x} - \boldsymbol{y}|), \quad \boldsymbol{x} \neq \boldsymbol{y}, \tag{5.3.49}$$

where v_n and w_n are polynomials to be determined. Note that Ψ_0 is also of the form (5.3.49), with $w_0 = 0$ and $v_0 = c$. Substituting (5.3.49) into (5.3.46) and equating the coefficients of $K_0(\alpha_0 r)$ and $K_1(\alpha_0 r)$ (where $r = |\boldsymbol{x} - \boldsymbol{y}|$), we find that v_n and w_n must solve the sequence of systems of ordinary differential equations, for $n = 1, \ldots, N-1$,

$$-v_n'' - \frac{1}{r}v_n' + 2\alpha_0 w_m' + (\alpha_n^2 - \alpha_0^2)v_n = \alpha_n^2 v_{n-1}, \tag{5.3.50}$$

$$2\alpha_0 v_n' - w_n'' + \frac{1}{r}w_n' - \frac{1}{r^2}w_n + (\alpha_n^2 - \alpha_0^2)w_n = \alpha_n^2 w_{n-1}. \tag{5.3.51}$$

In [12], only the situation where all time steps are equal is considered. In this case, $\alpha_0 = \ldots = \alpha_{N-1} =: \alpha$, and hence the last term on the left hand side of (5.3.50) and (5.3.51) disappears for all n. This leaves us with the same system of equations as [12, (3.5)], from which it is deduced that v_n and w_n take the form

$$v_n(r) = \sum_{k=0}^{\left[\frac{n}{2}\right]} a_{n,2k} r^{2k}, \quad w_n(r) = \sum_{k=0}^{\left[\frac{n-1}{2}\right]} a_{n,2k+1} r^{2k+1}, \tag{5.3.52}$$

where $[\cdot]$ denotes the integer part of a number, ie.

$$\left[\frac{j}{2}\right] = \begin{cases} \frac{j}{2}, & \text{if } j \text{ is even,} \\ \frac{j-1}{2}, & \text{if } j \text{ is odd.} \end{cases}$$

It can be shown that (5.3.52) satisfies (5.3.50) and (5.3.51) if and only if the

coefficients $a_{n,k}$ satisfy

$$a_{n,n} = \frac{\alpha}{2n} a_{n-1,n-1}, \quad n = 1, \ldots, N, \tag{5.3.53}$$

$$a_{n,k} = \frac{1}{2\alpha k}\left\{4\left[\frac{k+1}{2}\right]^2 a_{n,k+1} + \alpha^2 a_{n-1,k-1}\right\}, \quad k = n-1, \ldots, 1. \tag{5.3.54}$$

Provided the set $\{a_{n,0} : n = 0, \ldots, N-1\}$ is stipulated, then the set $\{a_{n,k} : n = 1, \ldots, N-1,\ k = 1, \ldots, n\}$ is uniquely determined from the relations (5.3.53), (5.3.54). For each n, $a_{n,0}$ is the coefficient of $K_0(\alpha|\boldsymbol{x} - \boldsymbol{y}|)$ in the formula for Ψ_n (5.3.49). In fact,

$$\Psi_n(\boldsymbol{x}, \boldsymbol{y}) = a_{n,0} K_0(\alpha|\boldsymbol{x} - \boldsymbol{y}|) + \text{smoother terms.} \tag{5.3.55}$$

As $(-\Delta_x + \alpha^2)K_0(\alpha|\boldsymbol{x} - \boldsymbol{y}|) = 0$, $\boldsymbol{x} \neq \boldsymbol{y}$, it is clear that $a_{n,0}$ can be arbitrarily chosen. Chapko and Kress choose $a_{n,0} = 1$, $n = 0, \ldots, N-1$.

Suppose however that the time steps are not equal, and in particular consider the situation where $\alpha_n \neq \alpha_0$ for all $n = 1, \ldots, N-1$. In this case (5.3.51) is clearly satisfied if $w_n = 0$ for all $n = 1, \ldots, N-1$, and v_n is a constant function for each n. In this case (5.3.50) reduces to

$$(\alpha_n^2 - \alpha_0^2)v_n = \alpha_n^2 v_{n-1}, \quad n = 1, \ldots, N-1,$$

and with $v_0 = c$ this gives us the general formula

$$v_n = c\prod_{j=1}^{n} \frac{\alpha_j^2}{\alpha_j^2 - \alpha_0^2}, \quad n = 1, \ldots, N-1. \tag{5.3.56}$$

So, for a variable time step, or at least one for which $h_n \neq h_0$ for all $n = 1, \ldots, N-1$, it is clear that a singular solution of (5.3.46) can be constructed, and it is also of a much simpler form than the solution constructed in [12] for a fixed time step. We can take advantage of this by specifically choosing all timesteps after the first to be different from the first, thus simplifying the form of (5.3.49). We give an example of how this could be done in Example 5.3.1 below.

Example 5.3.1 *Choose $h_0 = T/2N$, and set $h_1 = \ldots = h_{N-1} =: h$. To satisfy $\sum_{n=0}^{N-1} h_n = T$, we then need to take $h = ((1 - 1/2N)T)/(N-1)$, and then*

$$t_n = \frac{T}{2N}\left(1 + \frac{n(2N-1)}{N-1}\right), \quad n = 0, \ldots, N-1.$$

This then gives the system (5.3.43) with

$$\alpha_0 = \sqrt{\frac{2N}{T}}, \text{ and } \alpha_n = \sqrt{\frac{2N(N-1)}{T(2N-1)}}, \quad n = 1, \ldots, N-1.$$

It can then easily be shown that for all $j \neq 0$,

$$\frac{\alpha_j^2}{\alpha_j^2 - \alpha_0^2} = \frac{1-N}{N},$$

and so using (5.3.49) and (5.3.56), a singular solution of (5.3.46) is then given by

$$\Psi_n = c\left(\frac{1-N}{N}\right)^n K_0(\alpha_0|\boldsymbol{x} - \boldsymbol{y}|), \quad n = 0, \ldots, N-1. \tag{5.3.57}$$

As we shall show in §5.3.2, having the solution of (5.3.46) in this form significantly simplifies the integral equations which arise when seeking a solution which also satisfies the boundary condition (5.3.47).

5.3.2 Satisfying the boundary conditions

We now use the fundamental solutions constructed in §5.3.1 to solve the system of boundary value problems (5.3.46)–(5.3.47). We begin by defining

$$V_n(\boldsymbol{x}) := -\frac{1}{\pi}\sum_{m=0}^{n}\int_\Gamma q_m(\boldsymbol{y})\Psi_{n-m}(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}), \quad \boldsymbol{x} \in \mathbb{R}^2 \setminus \Gamma, \tag{5.3.58}$$

$$W_n(\boldsymbol{x}) := \frac{1}{\pi}\sum_{m=0}^{n}\int_\Gamma q_m(\boldsymbol{y})\frac{\partial}{\partial n(\boldsymbol{y})}\Psi_{n-m}(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}), \quad \boldsymbol{x} \in \mathbb{R}^2 \setminus \Gamma, \tag{5.3.59}$$

for $n = 0, \ldots, N-1$, where $q_n \in \mathcal{C}(\Gamma)$, and $\partial/\partial n(\boldsymbol{y})$ denotes differentiation with respect to the unit outward normal. Using the linearity of the differential and integral operators, it is clear that as the sequence Ψ_n, $n = 0, \ldots, N-1$, satisfies (5.3.46) then so do the sequences V_n, W_n, $n = 0, \ldots, N-1$. It just remains to satisfy the boundary condition (5.3.47).

In this section, we shall suppose that Ω is an unbounded domain, but note that similar results hold if Ω is bounded (although in that case the constants in the jump relations will change). First, in order that we can easily compare this approach with that of Lubich and Schneider (as described in §5.2.2), we consider the sequence V_n, and we shall assume that the time steps are all equal (to h) in which case Ψ_n is given as in [12]. The sequence V_n will then satisfy the boundary condition (5.3.47) if and only if its densities satisfy the sequence of first kind integral equations

$$-\frac{1}{\pi}\int_\Gamma q_n(\boldsymbol{y})\Psi_0(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}) = g_n(\boldsymbol{x}) + \frac{1}{\pi}\sum_{m=0}^{n-1}\int_\Gamma q_m(\boldsymbol{y})\Psi_{n-m}(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}), \quad \boldsymbol{x}\in\Gamma, \tag{5.3.60}$$

for $n = 0, \ldots, N-1$. Now (5.3.60) is virtually identical to the sequence of first kind integral equations (5.2.39) that must be solved in Lubich and Schneider's method. Taking $\delta(\xi) = 1 - \xi$ in (5.2.37) (as we are using the backward Euler method), it is immediately apparent that, with $\tilde{\Psi}$ representing the kernels of the integrals in (5.2.36),

$$\tilde{\Psi}_0(\boldsymbol{x},\boldsymbol{y}) = K_0\left(\frac{1}{\sqrt{h}}|\boldsymbol{x}-\boldsymbol{y}|\right) = \Psi_0(\boldsymbol{x},\boldsymbol{y}).$$

More generally, it is proved in [12] that $\Psi_n - \Psi_{n-1} = \tilde{\Psi}_n$. This confirms the assertion in Theorem 5.2.3 that the two methods do in fact lead to the same solution.

In practice, we will actually solve the system (5.3.46)–(5.3.47) using the sequence W_n (5.3.59), as this leads to a sequence of second kind integral equations, and the error analysis of Chapter 3 (which we shall employ in §5.4) is better developed for second kind integral equations than for first kind integral equations.

First, we assume that the time steps are all equal (to h). Using the classical

jump relations for the logarithmic potentials (see eg. [49, Theorem 6.13]), we then have that the sequence W_n satisfies the boundary condition (5.3.47) if and only if its densities satisfy the sequence of second kind integral equations

$$q_n(\boldsymbol{x}) + \frac{1}{\pi}\int_\Gamma q_n(\boldsymbol{y})\frac{\partial}{\partial n(\boldsymbol{y})}\Psi_0(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}) = g_n(\boldsymbol{x}) - \sum_{m=0}^{n-1} q_m(\boldsymbol{x}) - \frac{1}{\pi}\sum_{m=0}^{n-1}\int_\Gamma q_m(\boldsymbol{y})\frac{\partial}{\partial n(\boldsymbol{y})}\Psi_{n-m}(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}), \quad \boldsymbol{x}\in\Gamma, \tag{5.3.61}$$

for $n = 0,\ldots,N-1$. Note that for (5.3.61) (and also (5.3.60)), on each time level the integral operator to be inverted remains the same and only the right hand side changes.

If instead we use a variable timestepping scheme, which satisfies $h_n \neq h_0$, for all $n = 1,\ldots,N-1$, then the densities will now satisfy a slightly different sequence of second kind integral equations. This is because the coefficients $a_{n,0}$ of $K_0(\alpha_0|\boldsymbol{x}-\boldsymbol{y}|)$ in the formula for v_n in (5.3.49) can no longer be forced to be equal to one, as is the case in [12]. Recalling (5.3.55), these coefficients of $K_0(\alpha_0|\boldsymbol{x}-\boldsymbol{y}|)$ correspond to the coefficients of the logarithmic singularity in Ψ_n, and hence as they vary, so the jump conditions as $\boldsymbol{x}\to\Gamma$ change.

For example, using the timestepping scheme of Example 5.3.1, so that Ψ_n is given by (5.3.57), and choosing the constant $c = 1$ in (5.3.57), we then have a similar sequence of integral equations to (5.3.61) with the same left hand side, but the right hand side is now given by

$$g_n(\boldsymbol{x}) - \sum_{m=0}^{n-1}\left(\frac{1-N}{N}\right)^{n-m} q_m(\boldsymbol{x}) - \frac{1}{\pi}\sum_{m=0}^{n-1}\int_\Gamma q_m(\boldsymbol{y})\frac{\partial}{\partial n(\boldsymbol{y})}\Psi_{n-m}(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}). \tag{5.3.62}$$

We now consider the numerical solution of the sequence of second kind integral equations (5.3.61) (or the similar sequence with right hand side given by (5.3.62)).

Regardless of the timestepping scheme used, $\Psi_0(\boldsymbol{x},\boldsymbol{y})$ is up to a constant equal to the fundamental solution of the Helmholtz equation, and so for each n, the left hand side of (5.3.61) is of exactly the same form as the left hand side of the second kind integral equations (3.1.5) that are solved in Chapter 3. In [12], it

is proposed to solve (5.3.61), using the Nyström method (3.1.18), with a splitting method similar to (3.1.16) to deal with the logarithmic singularities. This leads to problems as described in §3.1 when α_0 in (5.3.43) is large, ie. when the time step h_0 is small. As h_0 is decreased, one needs to use more and more boundary nodes for the solution of the integral equations, and the method rapidly becomes infeasible. Numerical results demonstrating this are shown in §5.5.

As an alternative, we propose to solve the integral equations (5.3.61) using the collocation method of Chapter 3. Numerical results demonstrating how this method is superior to the splitting method described above can also be found in §5.5.

When solving (5.3.61), care must be taken to evaluate the weakly singular integrals on the right hand side accurately. If a constant timestep is used, each Ψ_n will be of the form (5.3.49), with v_n and w_n as in (5.3.52). Thus each $\Psi_n(\boldsymbol{x}, \boldsymbol{y})$ has a weak singularity at $\boldsymbol{x} = \boldsymbol{y}$, and will be peaked near $\boldsymbol{x} = \boldsymbol{y}$ when α_0 is large. The transformation and graded mesh scheme of §3.4 is certainly a good way to evaluate these integrals, and this is how we compute the right hand side for the numerical experiments in §5.5. Note however that for each n one must first evaluate the coefficients $a_{j,k}$ for the polynomials v_n and w_n, and then sample each Ψ_n at the points of the graded mesh. As this must be carried out separately for each value of n, this will be very expensive, and indeed it is the setting up of the right hand side of each equation rather than the actual solve that takes up the bulk of the computing time.

However, if we use the variable time stepping scheme of Example 5.3.1, then each Ψ_n is just a constant multiple of Ψ_0. Hence the graded mesh need only be applied once, to Ψ_0, and the integrals on the right hand side can be deduced easily from this. We conjecture that this method should greatly reduce the computing cost of the scheme, and this may be an interesting area for future research.

5.4 Error analysis

In this section, we shall discuss an error analysis for the method of §5.3 for solving the initial boundary value problem (5.1.1), (5.1.7) and (5.1.8) by turning it into the spatial boundary value problem (5.3.46)–(5.3.47), and then solving this system by means of the solution of the equivalent sequence of boundary integral

equations

$$q_n(\boldsymbol{x}) + \frac{1}{\pi}\int_\Gamma q_n(\boldsymbol{y})\frac{\partial}{\partial n(\boldsymbol{y})}\Psi_0(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}) = g_n(\boldsymbol{x}) - \sum_{m=0}^{n-1}\left(\frac{1-N}{N}\right)^{n-m} q_m(\boldsymbol{x}) - \frac{1}{\pi}\sum_{m=0}^{n-1}\int_\Gamma q_m(\boldsymbol{y})\frac{\partial}{\partial n(\boldsymbol{y})}\Psi_{n-m}(\boldsymbol{x},\boldsymbol{y})ds(\boldsymbol{y}), \tag{5.4.63}$$

for $\boldsymbol{x} \in \Gamma$, $n = 0, \ldots, N-1$, using the method of Chapter 3. This corresponds to the use of the variable timestepping scheme of example 5.3.1, so that Ψ_n is given by (5.3.57).

Assuming that Γ is parametrised by a 2π-periodic function $\boldsymbol{\gamma} : \mathbb{R} \mapsto \Gamma$ which satisfies (1.1.5)–(1.1.7), we can write (5.4.63) in parametric form as

$$q_n(t) + G_0 q_n(t) = g_n(t) - \sum_{m=0}^{n-1}\left\{\left(\frac{1-N}{N}\right)^{n-m} q_m(t) + G_{n-m}q_m(t)\right\}, \quad t \in [0, 2\pi], \tag{5.4.64}$$

for $n = 0, \ldots, N-1$, where (with a slight abuse of notation) q_n and g_n now denote $q_n \circ \boldsymbol{\gamma}$ and $g_n \circ \boldsymbol{\gamma}$ respectively, and for any function $\psi \in \mathcal{C}(\Gamma)$,

$$G_n\psi(t) = \frac{1}{\pi}\int_0^{2\pi}\left\{\frac{\partial}{\partial n(\boldsymbol{y})}\Psi_n(\boldsymbol{x},\boldsymbol{y})\Big|_{\boldsymbol{x}=\boldsymbol{\gamma}(t),\boldsymbol{y}=\boldsymbol{\gamma}(\tau)}\right\}\psi(\tau)|\boldsymbol{\gamma}'(\tau)|d\tau, \quad t \in [0, 2\pi], \tag{5.4.65}$$

with Ψ_n as in (5.3.57). We can then write the whole sequence (5.4.64) in matrix form as

$$(\mathcal{L} + \mathcal{G})\boldsymbol{q} = \boldsymbol{g}, \tag{5.4.66}$$

where $\boldsymbol{q} = (q_0, \ldots, q_{N-1})^T$ and $\boldsymbol{g} = (g_0, \ldots, g_{N-1})^T$. Here $\mathcal{L} = \{l_{i,j}\}$ is the lower triangular matrix defined by

$$l_{i,j} = \left\{\begin{array}{ll} I & i \geq j \\ 0 & i < j \end{array}\right\}, \quad i, j = 1, \ldots, N,$$

I is the identity operator, and $\mathcal{G}$ is the matrix of operators

$$\mathcal{G} = \begin{pmatrix} G_0 & 0 & 0 & \cdot & 0 \\ G_1 & G_0 & 0 & \cdot & 0 \\ G_2 & G_1 & G_0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ G_{N-1} & G_{N-2} & G_{N-3} & \cdot & G_0 \end{pmatrix}.$$

To solve (5.4.66) we then use the collocation method of Chapter 3, with $2M$ collocation points (ie. with M taking the role of n in Chapter 3). Recalling the definitions of T_M (3.1.9) and $t_j^{(M)}$ (3.1.11), we then seek a function

$$\tilde{\boldsymbol{q}}_M = (\tilde{q}_0, \ldots, \tilde{q}_{N-1})^T \in (T_M)^N,$$

such that

$$(\mathcal{L} + \mathcal{G})\tilde{\boldsymbol{q}}_M \left(t_j^{(M)}\right) = \boldsymbol{g}\left(t_j^{(M)}\right), \quad j = 0, \ldots, 2M-1, \tag{5.4.67}$$

where

$$\tilde{\boldsymbol{q}}_M \left(t_j^{(M)}\right) = \left(\tilde{q}_0(t_j^{(M)}), \ldots, \tilde{q}_{N-1}(t_j^{(M)})\right)^T,$$

and

$$\boldsymbol{g}\left(t_j^{(M)}\right) = \left(g_0(t_j^{(M)}), \ldots, g_{N-1}(t_j^{(M)})\right)^T.$$

For simplicity, here we assume that the integrals

$$G_i\tilde{q}_j\left(t_k^{(M)}\right), \quad i,j = 0, \ldots, N-1, \quad k = 0, \ldots, 2M-1, \tag{5.4.68}$$

are all evaluated exactly. In practice the graded mesh scheme of Chapter 3 is used. Note that with Ψ_n given by (5.3.57), all of the operators G_n are just constant multiples of the operator K_α of Chapter 3, and so they have the same blow up near zero and exponential decay away from zero. Hence the error bounds of Chapter 3 can be used for the approximation of (5.4.68).

Having found $\tilde{\boldsymbol{q}}_M$, the approximate solution of the initial boundary value

problem (5.1.1), (5.1.7) and (5.1.8) is given for each $n = 0, \ldots, N-1$ by

$$u_{n,M}(\boldsymbol{x}) = \sum_{j=0}^{n} \frac{1}{\pi} \int_{\Gamma} \frac{\partial}{\partial n(\boldsymbol{y})} \Psi_{n-j}(\boldsymbol{x}, \boldsymbol{y}) \tilde{q}_j(\boldsymbol{y}) ds(\boldsymbol{y}), \quad \boldsymbol{x} \in \Omega.$$

Defining u to be the exact solution of (5.1.1), (5.1.7) and (5.1.8), and u_n to be the exact solution of (5.3.43)–(5.3.45) (ie. the semidiscrete (in time) approximation to u), an error bound for the fully discrete method can be acheived by considering the temporal and spatial errors separately, as

$$\begin{aligned} \|u - u_{n,M}\|_{L_2(\Omega)} &\leq \|u - u_n\|_{L_2(\Omega)} + \|u_n - u_{n,M}\|_{L_2(\Omega)} && (5.4.69) \\ &=: e_1 + e_2. && (5.4.70) \end{aligned}$$

Using the link between the methods of §5.2 and §5.3, as proved in Theorem 5.2.3 and demonstrated in §5.3, it is clear that we can bound e_1, using (5.2.41) and the fact that $\|\cdot\|_{L_2(\Omega)} \leq \|\cdot\|_{H^1(\Omega)}$. Alternatively, we could use the error bound proved in [12, Theorem 5.1],

$$\sup_{n=0,\ldots,N-1} \|u_n - u(\cdot, (n+1)h)\|_{\infty,\Omega} \leq \frac{h}{2} \left\| \frac{\partial^2 g}{\partial t^2} \right\|_{\infty,\Gamma\times[0,T]}, \qquad (5.4.71)$$

and the fact that $\|\cdot\|_{L_2(\Omega)} \leq C\|\cdot\|_{L_\infty(\Omega)}$. Either way, bounding e_1 is not a problem.

The problem of bounding e_2 is more difficult. We have managed to establish an error bound showing that e_2 is superalgebraically small with respect to the number of space steps M, but the constants in this error bound depend in an undetermined way on the number of time steps N. We have so far been unable to prove a bound for e_2 with constants independent of N, and so we do not discuss this any further here. This may be an interesting area for future research.

5.5 Numerical results

In this section we use the same numerical example that appears in [12], although note that there it is solved using a single layer potential representation whereas here we use the double layer potential representation (5.3.59). We solve the heat

equation (5.1.1) with $c = 1$, $T = 1$, on the two-dimensional domain external to the boundary curve

$$\Gamma = \{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} = \boldsymbol{x}(s) := (0.2\cos s, 0.4\sin s - 0.3\sin^2 s), \quad 0 \le s \le 2\pi\}, \tag{5.5.72}$$

which is shown in Figure 5-1 below, with boundary data given by the restriction

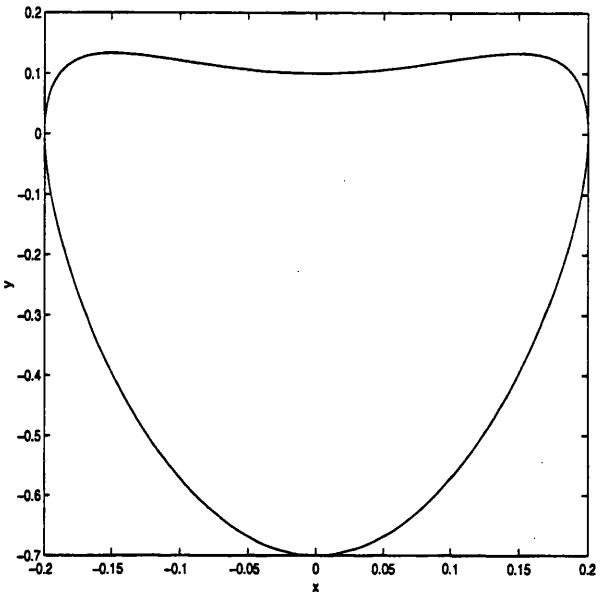

Figure 5-1: Boundary curve (5.5.72).

of the fundamental solution

$$u(\boldsymbol{x}, t) = \frac{1}{4\pi t} \exp\left(-\frac{|\boldsymbol{x}|^2}{4t}\right), \quad |\boldsymbol{x}| > 0, \quad t > 0,$$

to the boundary Γ, and a homogeneous initial condition. To solve this problem we discretise in time using the backward Euler method with a fixed time step $h = T/N$ to get the sequence of spatial boundary value problems (5.3.43)–(5.3.45), and we reformulate these as the sequence of integral equations (5.3.61).

First, we solve each of the integral equations (5.3.61), $n = 0, \ldots, N-1$, using the Nyström method (3.1.18) (with the splitting (3.1.16) to deal with the logarithmic singularities) as suggested in [12]. Table 5.1 shows the error at $T = 1$ between the exact solution and the numerical solution at the two points $\boldsymbol{x} =$

$(0.3, 0)$ and $\boldsymbol{x} = (0.6, 0)$. Here N is the number of time steps, and M is the number of boundary nodes used in the Nyström method (3.1.18).

The linear convergence with respect to the time step expected of the backward Euler method can initially be seen, but when the time step gets small the constant $C(\alpha)$ in the error bound (3.1.19) for the solution of each integral equation gets so large that the error bound is useless, even though the method still seems to be converging superalgebraically with respect to M. Table 3.1 in Chapter 3 demonstrates that all convergence may disappear if the timestep becomes too small.

Next, we use the discrete collocation method (3.5.52) to solve the integral equations (5.3.61). Table 5.2 again shows the error at $T = 1$ between the exact solution and the numerical solution at the two points $\boldsymbol{x} = (0.3, 0)$ and $\boldsymbol{x} = (0.6, 0)$. Note that here we chose the parameter m in (3.5.52) (the number of quadrature points used in the graded mesh) to be sufficiently large that the error would be almost entirely dependent just on N (the number of time steps) and n (the number of basis functions). The exact values used for the graded mesh were $N = 25$ (N is the number of points in $[0, 1]$), $q1 = 2$, $q2 = \sqrt{N}$, and $r = 10$.

Provided n is sufficiently large, the linear convergence with respect to the time step can be seen, although the convergence is a little erratic. It is clear that the method continues to work well as the time step gets smaller. Whereas the errors in Table 5.1 are enormous for $N = 80$ (or equivalently $\alpha \approx 9$ in (5.3.43)), the method in Table 5.2 continues to work well as $h \to 0$.

	$x = (0.3, 0)$			$x = (0.6, 0)$		
N	M=16	M=32	M=48	M=16	M=32	M=48
10	1.964e-3	1.912e-3	1.902e-3	5.708e-3	5.671e-3	5.663e-3
20	9.284e-4	8.947e-4	8.880e-4	2.575e-3	2.537e-3	2.530e-3
40	4.230e+1	2.961e-4	2.912e-4	1.920e-1	7.181e-4	7.098e-4
80	1.9e+68	1.9e+45	1.1e+31	5.7e+65	1.7e+42	5.6e+27

Table 5.1: Errors using the "splitting" method to solve (5.3.61).

	$x = (0.3, 0)$			$x = (0.6, 0)$		
N	n=16	n=32	n=48	n=16	n=32	n=48
10	4.629e-4	1.708e-3	1.708e-3	5.450e-3	5.445e-3	5.445e-3
20	6.029e-4	7.834e-4	7.824e-4	2.399e-3	2.400e-3	2.400e-3
40	1.230e-3	2.567e-4	2.558e-4	6.531e-4	6.552e-4	6.552e-4
80	1.471e-3	4.131e-5	4.061e-5	6.248e-5	6.279e-5	6.271e-5

Table 5.2: Errors using the discrete collocation method of Chapter 3 to solve (5.3.61).

Chapter 6

Applying the domain embedding method to parabolic PDEs

6.1 Introduction

In this chapter, we present some numerical experiments demonstrating how the ideas of Chapters 2–4 can be applied to the solution of parabolic PDEs of the form

$$\frac{1}{c}\frac{\partial u}{\partial t} = \Delta u + f(u), \quad \text{in } \Omega \times (0,T], \tag{6.1.1}$$

subject to the initial condition

$$u(\cdot,0) = u_0, \quad \text{in } \Omega, \tag{6.1.2}$$

and for simplicity we consider here only the Dirichlet boundary condition

$$u = g, \quad \text{on } \Gamma \times [0,T]. \tag{6.1.3}$$

Here $c, T > 0$, f is some (possibly nonlinear) reaction term, and $\Omega \subset \mathbb{R}^2$ is a bounded and simply connected domain with boundary Γ satisfying (1.1.5) and (1.1.7). Note that due to the inhomogeneity $f(u)$, we cannot in general use the approach in Chapter 5 for this problem.

In this chapter, we solve (6.1.1)–(6.1.3) using the domain embedding procedure outlined in Chapter 1. For clarity we repeat this procedure here. We choose

the time points $0 = t_0 < t_1 < \ldots < t_M$, and then discretise (6.1.1) using linearly implicit backward Euler timestepping to get

$$\frac{u_{n+1} - u_n}{c\delta t_n} = \Delta u_{n+1} + f(u_n), \quad \text{in } \Omega, \quad n = 0, \ldots, M-1, \tag{6.1.4}$$

where $\delta t_n = t_{n+1} - t_n$, and $u_n = u(\cdot, t_n)$. Rearranging, we get the sequence of inhomogeneous Helmholtz equations,

$$-\Delta u_{n+1} + \frac{1}{c\delta t_n} u_{n+1} = \frac{1}{c\delta t_n} u_n + f(u_n), \quad \text{in } \Omega, \quad n = 0, \ldots, M-1, \tag{6.1.5}$$

with u_0 given by (6.1.2), subject to the boundary conditions

$$u_{n+1} = g_{n+1}, \quad \text{on } \Gamma, \quad n = 0, \ldots, M-1, \tag{6.1.6}$$

where $g_n = g(\cdot, t_n)$. Each of the problems (6.1.5)–(6.1.6) can then be solved using the domain embedding method of Chapters 2–4. For each n, we begin by using the method of Chapter 2 to compute a particular solution $(u_p)_{n+1}$ to (6.1.5), and then we use bilinear interpolation to determine the values of $(u_p)_{n+1}$ on Γ. We then solve the homogeneous Helmholtz-type problem

$$\begin{aligned} -\Delta v_{n+1} + \frac{1}{c\delta t_n} v_{n+1} &= 0, \quad \text{in } \Omega, \\ v_{n+1} &= g_{n+1} - (u_p)_{n+1}, \quad \text{on } \Gamma, \end{aligned}$$

using the collocation method of Chapter 3, and then we use the method of Chapter 4 to determine the values of v_{n+1} throughout Ω. For each n, the solution of (6.1.5)–(6.1.6) is then given by $u_{n+1} = v_{n+1} + (u_p)_{n+1}$, and this gives us the data for the right hand side of (6.1.5) at the next time level. The timestepping may be continued until a steady state is reached, ie. t_M need not be fixed in advance.

In the case $f \equiv 0$ (ie. the heat equation) this method is directly comparable with the methods of [56] and [12], as described in Chapter 5. The main advantages the domain embedding approach has over these methods are that it does not require a homogeneous initial condition, and also it avoids convolution in time and thus has storage advantages, as there is no need to store all solutions from previous time levels.

If $f \not\equiv 0$ however, and in particular if f is nonlinear, then the approaches

of both [56] and [12] are inapplicable, and so the method described here is a genuine extension of the methods of [56] and [12] to these more general parabolic problems.

To extend the time discretisation scheme of [56] to the solution of (6.1.1)–(6.1.3) would require knowledge of the the Laplace transform of the fundamental solution of (6.1.1). It can be shown (see for example [19, Theorem 8, Chapter 1]) that any linear parabolic PDE of the form

$$Lu \equiv \sum_{i,j=1}^{n} a_{i,j}(\boldsymbol{x},t)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{n} b_i(\boldsymbol{x},t)\frac{\partial u}{\partial x_i} + c(\boldsymbol{x},t)u - \frac{\partial u}{\partial t} = 0, \quad \text{in}\,\Omega \times (0,T], \tag{6.1.7}$$

has fundamental solution given by

$$\begin{aligned} G(\boldsymbol{x},t;\boldsymbol{y},\tau) \;=\; & \frac{1}{4\pi(t-\tau)} \exp\left(\frac{-|\boldsymbol{x}-\boldsymbol{y}|^2}{4(t-\tau)}\right) \\ & + \int_\tau^t \int_\Omega \frac{1}{4\pi(t-\sigma)} \exp\left(\frac{-|\boldsymbol{x}-\boldsymbol{\eta}|^2}{4(t-\sigma)}\right) \Phi(\boldsymbol{\eta},\sigma,\boldsymbol{y},\tau)\,d\boldsymbol{\eta}\,d\sigma, \end{aligned} \tag{6.1.8}$$

where Φ is determined by using the condition that $LG = 0$. However, this does not cover the case of nonlinear PDEs, and even for many linear PDEs the formula (6.1.8) involves a domain integral, which may make the evaluation of the Laplace transform difficult.

In general it will also be difficult to extend the method of [12] to the solution of (6.1.1)–(6.1.3). If the reaction term f is linear, then it may be possible to determine a formula for a singular solution of (6.1.5)–(6.1.6), but for a general nonlinear f this will be hard, if not impossible.

An outline of this chapter is as follows.

- In §6.2 we use the domain embedding method to solve the Dirichlet problem for the heat equation over a fixed time domain, ie. we solve (6.1.1)–(6.1.3) with $f \equiv 0$ for fixed $T > 0$.

- In §6.3 we solve the Dirichlet problem for Fisher's equation ie. (6.1.1)–(6.1.3) with $f(u) \equiv ku(1-u)$, $k \geq 0$ constant. In this case, the linearly implicit backward Euler timestepping approach is no longer stable for all

time steps. As well as solving the problem for fixed $T > 0$, we also compute steady state solutions by fixing the time step and then continuing timestepping until an approximate steady state is reached.

In both sections we will briefly discuss the problem under investigation, and then present numerical results demonstrating the good performance of the domain embedding method in determining accurate solutions to these problems.

6.2 Solving the heat equation by domain embedding.

We present two numerical experiments in this section. In each case we solve the heat equation (6.1.1) with $c = 1$ and $f \equiv 0$.

Example 6.2.1 *As a first example, we consider the very simple case where Ω is the unit circle, and where the Dirichlet data in (6.1.3) is given by $g \equiv 0$. Taking the initial condition in (6.1.2) as $u_0(\boldsymbol{x}) = J_0(c_0|\boldsymbol{x}|)$, where J_0 is the first kind Bessel function of order zero and $c_0 \approx 2.4048$ is the first zero of J_0 (ie. $J_0(c_0) = 0$), it can then be shown that the exact solution is given by*

$$u(\boldsymbol{x}, t) = J_0(c_0|\boldsymbol{x}|)e^{-c_0^2 t}. \tag{6.2.9}$$

Example 6.2.2 *As a second example, we take Ω to be the ellipse with boundary*

$$\Gamma = \{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} = \boldsymbol{x}(s) := (3\cos s, \sin s),\ 0 \le s \le 2\pi\}, \tag{6.2.10}$$

and we then solve the problem with Dirichlet data given by (6.1.3) with

$$g(\boldsymbol{x}, t) = \frac{1}{4\pi(t + 0.1)} \exp\left(-\frac{|\boldsymbol{x}|^2}{4(t + 0.1)}\right), \quad \boldsymbol{x} \in \Gamma, \quad t \in [0, T],$$

and initial condition given by (6.1.2) with

$$u_0(\boldsymbol{x}) = \frac{5}{2\pi} \exp\left(-\frac{5|\boldsymbol{x}|^2}{2}\right), \quad \boldsymbol{x} \in \Omega,$$

in which case the exact solution is given by

$$u(\boldsymbol{x},t) = \frac{1}{4\pi(t+0.1)} \exp\left(-\frac{|\boldsymbol{x}|^2}{4(t+0.1)}\right), \quad \boldsymbol{x} \in \Omega, \quad t \in [0,T]. \tag{6.2.11}$$

For each example, we solve (6.1.1)–(6.1.3) for $T = 0.01$, $T = 0.1$ and $T = 1.0$, in order to demonstrate the robustness of the method for either large or small timesteps. For each value of T we discretise (6.1.1) in time using backward Euler timestepping with a fixed time step $\delta t = T/M$, in which case we would expect the convergence to be linear with respect to δt. As $f \equiv 0$ this method is fully implicit, and so the method is unconditionally stable (see for example [81, Chapter 6]).

Having discretised in time we then solve the resulting sequence of inhomogeneous Helmholtz problems (6.1.5)–(6.1.6) using the domain embedding method, as described in §6.1. The errors between the exact and approximate solutions for Examples 6.2.1 and 6.2.2 at the points $(0,0)$ and $(0,0.3125)$ are given in Tables 6.1 and 6.2 respectively.

T	n	n_c	N	M	error(0,0)	EOC	error(0,0.3125)	EOC
0.01	31	14	4	1	1.959e-3	-1.9	1.630e-3	-1.9
	63	17	16	4	5.096e-4	-2.0	4.275e-4	-2.1
	127	20	64	16	1.286e-4	-2.0	9.708e-5	-2.3
	255	23	256	64	3.227e-5		1.938e-5	
0.1	31	14	4	1	7.404e-2	-1.7	6.365e-2	-1.8
	63	17	16	4	2.203e-2	-1.9	1.881e-2	-2.0
	127	20	64	16	5.713e-3	-2.0	4.798e-3	-2.1
	255	23	256	64	1.404e-3		1.144e-3	
1.0	31	14	4	1	1.451e-1	-2.5	1.253e-1	-2.5
	63	17	16	4	2.494e-2	-2.6	2.154e-2	-2.6
	127	20	64	16	4.095e-3	-2.3	3.536e-3	-2.3
	255	23	256	64	8.577e-4		7.407e-4	

Table 6.1: Errors for Example 6.2.1.

The parameters n, n_c, and N for the solution of each problem (6.1.5)–(6.1.6) denote the number of grid points of the artificial domain R_h in each direction (see (2.3.13)), the number of collocation points for the boundary integral equations (see (3.1.11)) and the number of quadrature points in $[0,1]$ used to compute

T	n	n_c	N	M	error(0,0)	EOC	error(0,0.3125)	EOC
0.01	31	14	4	1	1.103e-2	-1.8	5.276e-3	-1.8
	63	17	16	4	3.152e-3	-1.9	1.498e-3	-1.9
	127	20	64	16	8.188e-4	-2.0	3.890e-4	-2.0
	255	23	256	64	2.032e-4		9.429e-5	
0.1	31	14	4	1	8.815e-2	-1.7	5.849e-2	-1.7
	63	17	16	4	2.641e-2	-1.9	1.777e-2	-1.7
	127	20	64	16	7.185e-3	-1.9	5.454e-3	-1.8
	255	23	256	64	1.868e-3		1.547e-3	
1.0	31	14	4	1	8.761e-2	-2.3	7.294e-2	-2.2
	63	17	16	4	1.783e-2	-2.1	1.571e-2	-2.1
	127	20	64	16	4.105e-3	-2.0	3.656e-3	-1.9
	255	23	256	64	1.043e-3		9.761e-4	

Table 6.2: Errors for Example 6.2.2.

the collocation matrix entries (see (3.4.40)) respectively, exactly as for the numerical experiments in §4.4. With n, n_c and N chosen as in Tables 6.1 and 6.2 we would then expect the estimated order of convergence (EOC) (computed using the formula (2.6.57)) for the solution of each problem (6.1.5)–(6.1.6) to satisfy EOC ≈ -2 (see §4.4 for details). In order to maintain this convergence rate for the solution of the heat equation, and taking the linear convergence with respect to the time step into account, it is clear that as we double n we must then quadruple M. This is what we do in Tables 6.1 and 6.2.

The expected convergence rate can clearly be seen for both examples, and for all values of T, demonstrating the predicted convergence rate of the method, and also the robustness of the method even for a very small time step.

6.3 Solving Fisher's equation by domain embedding.

As mentioned in §6.2, it is well known (see eg. [81]) that the backward Euler time discretisation method is unconditionally stable. However, schemes which are not fully implicit, such as the linearly implicit backward Euler method (6.1.4) when $f \not\equiv 0$, will not have this property. The stability of such a scheme will often depend on the parameter $\mu = \delta t/h^2$, where δt is the time step, and h is the space

step. Timestepping schemes which are not fully implicit will only be stable if μ is sufficiently small.

For example, if we discretise the heat equation (corresponding to (6.1.1) with $f \equiv 0$) using explicit Euler timestepping, we get

$$\frac{u_{n+1} - u_n}{c\delta t_n} = \Delta u_n, \quad \text{in } \Omega, \quad n = 0, \ldots, M-1, \tag{6.3.12}$$

and this scheme will only be stable if (see [81, Chapter 6])

$$\mu := \frac{\delta t}{h^2} \leq \frac{1}{2c}.$$

For the linearly implicit backward Euler method applied to (6.1.1), the stability analysis will depend intimately on the properties of f and on the solution u. We do not present such a stability analysis here, but rather we shall just determine experimentally a value of μ which gives us a stable method, ie. for a fixed spatial grid we shall progessively decrease the timestep until we find a value of μ that gives good results.

We shall begin by seeking a solution of (6.1.1)–(6.1.3) at a fixed time, and this will enable us to determine a value of μ which gives us a stable method. Once such a value has been found, we shall then use this to seek steady states for the problem, by solving (6.1.5)–(6.1.6) repeatedly, for a fixed time step, until we have $\|u^{n+1} - u^n\|_\infty < tol$, for some predetermined tolerance level tol.

Our test problem is Fisher's equation, which is given by

$$\frac{\partial u}{\partial t} = \Delta u + ku(1-u), \quad \text{in } \Omega \times (0, T], \tag{6.3.13}$$

where $k \geq 0$ is a parameter which we shall vary. Writing $\boldsymbol{x} = (x_1, x_2)$, we supplement (6.3.13) with the initial condition

$$u(\boldsymbol{x}, 0) = J_0\left(c_0\sqrt{\frac{x_1^2}{9} + x_2^2}\right), \quad \text{in } \Omega, \tag{6.3.14}$$

where J_0 is the first kind Bessel function of order zero and $c_0 \approx 2.4048$ is the first

zero of J_0 as in §6.2, and also the Dirichlet boundary condition

$$u = 0, \quad \text{on } \Gamma \times [0,T]. \tag{6.3.15}$$

Throughout this section we shall take Ω to be the ellipse with boundary given by (6.2.10). Note that the initial condition (6.3.14) satisfies (6.3.15).

The existence and uniqueness of a solution to (6.3.13)–(6.3.15) follows from [10, Theorem 4.19], where a more general existence and uniqueness result for reaction diffusion problems of the form (6.1.1)–(6.1.3) under certain assumptions on f, Ω, u_0 and g is proved. By [10, Theorem 4.60], we also have that there exists a constant $\lambda > 0$ such that if $k < \lambda$ then zero is a stable steady state, and if $k > \lambda$ then zero is an unstable steady state. Moreover, by [10, Theorem 4.61] we have that if $k > \lambda$ then there exists at least one non-trivial non-negative steady state solution of (6.3.13)–(6.3.15) (ie. a non-trivial non-negative function v that satisfies $\Delta v + kv(1-v) = 0$ in Ω, $v = 0$ on Γ). In this section we shall attempt to verify these results numerically, and to compute such non-trivial steady states.

We solve (6.3.13)–(6.3.15) for various values of k, using linearly implicit backward Euler timestepping, leading to the sequence of inhomogeneous Helmholtz problems (6.1.5)–(6.1.6). We solve these using the domain embedding method as described in §6.1. For each value of k we begin by solving (6.3.13)–(6.3.15) at fixed time $T = 1$, using various timesteps, and then once we have determined a value of μ which gives us reasonable convergence we repeatedly solve (6.1.5)–(6.1.6) until we reach a steady state.

So, to begin with we solve (6.3.13)–(6.3.15) with $T = 1$ and $k = 1$, for various values of μ. The exact solution is unknown, so we take the "near exact" solution to be that computed with $n = 255$, $n_c = 23$, $N = 256$, and $M = 512$, where these parameters are as in §6.2. The errors between the "near exact" and approximate solutions at the points $(0,0)$ and $(0, 0.3125)$ are given in Table 6.3. As in §6.2 the estimated order of convergence (EOC) was computed using the formula (2.6.57), and for a stable method we would expect this to satisfy EOC ≈ -2.

Clearly the expected order of convergence is acheived for all of the different values of μ. We now seek a steady state. We solve (6.1.5)–(6.1.6) repeatedly with $n = 63$, $n_c = 17$, $N = 16$, and $\delta t = 0.125$ (correspondingly $\mu = 5.12$) until $\|u^{n+1} - u^n\|_\infty < 1e-6$. The values of $u^n(0,0)$ and $u^n(0, 0.3125)$ against time are

n	n_c	N	M	μ	error(0,0)	EOC	error(0,0.3125)	EOC
31	14	4	1	10.24	1.965e-1	-1.3	1.777e-1	-1.3
63	17	16	4		7.973e-2	-1.8	7.048e-2	-1.8
127	20	64	16		2.263e-2	-1.8	1.998e-2	-1.8
255	23	256	64		6.430e-3		5.720e-3	
31	14	4	2	5.12	1.363e-1	-1.7	1.210e-1	-1.7
63	17	16	8		4.303e-2	-2.0	3.788e-2	-2.0
127	20	64	32		1.113e-2	-1.9	9.800e-3	-1.9
255	23	256	128		3.030e-3		2.730e-3	
31	14	4	4	2.56	8.083e-2	-1.9	7.128e-2	-1.9
63	17	16	16		2.123e-2	-1.0	1.858e-2	-1.0
127	20	64	64		1.073e-2	-3.1	9.270e-3	-3.0
255	23	256	256		1.290e-3		1.160e-3	
31	14	4	8	1.28	4.083e-2	-2.4	3.568e-2	-2.4
63	17	16	32		7.930e-3	-4.6	6.730e-3	-5.4
127	20	64	128		3.300e-4		1.600e-4	

Table 6.3: Errors for solving (6.3.13)–(6.3.15) with $T = 1$ and $k = 1$.

plotted in Figures 6-1 and 6-2 respectively.

Convergence to the zero steady state can clearly be seen, indicating that for $k = 1$, zero is a stable steady state, and thus $1 < \lambda$, where λ is the critical value of k at which zero becomes unstable, as explained above.

We next solved the same problem with $k = 10$. As for $k = 1$ we began by determining a suitable value of μ such that we acheived good convergence to the solution at a fixed time T, and then we solved (6.1.5)–(6.1.6) repeatedly, this time with $n = 63$, $n_c = 17$, $N = 16$, and $\delta t = 0.0125$ (correspondingly $\mu = 0.512$) until $\|u^{n+1} - u^n\|_\infty < 1e - 6$. The values of $u^n(0,0)$ and $u^n(0,0.3125)$ against time are plotted in Figures 6-3 and 6-4 respectively.

Clearly we now have convergence to a non-trivial steady state, and this is shown in Figure 6-5. We can thus deduce that the critical value λ satisfies $1 < \lambda < 10$. Note that the elliptic domain Ω is the region in Figure 6-5 where the solution is non-zero, and the fictitious domain R is the whole region which is meshed.

In order to get a clearer idea of the exact value of λ we repeated the above procedure for $k = 5$ and $k = 3$. In each case we solved (6.1.5)–(6.1.6) repeatedly with $n = 63$, $n_c = 17$, $N = 16$, and $\delta t = 0.125$ (correspondingly $\mu = 5.12$) until

$||u^{n+1} - u^n||_\infty < 1e-6$.

For $k = 5$ we again got convergence to a non-trivial steady state. The values of $u^n(0,0)$ and $u^n(0,0.3125)$ against time are plotted in Figures 6-6 and 6-7 respectively, and the steady state solution is shown in Figure 6-8. From Figures 6-6 and 6-7 it is clear that the convergence to this steady state is much slower than for $k = 10$, suggesting that we are now much closer to the critical value of k. Also the uniform norm of the steady state seems to be decreasing as k decreases.

For $k = 3$ the solution converges to the trivial steady state. The values of $u^n(0,0)$ and $u^n(0,0.3125)$ against time are plotted in Figures 6-9 and 6-10 respectively. From Figures 6-9 and 6-10 it is clear that the convergence to this steady state is very slow indeed, and this suggests that λ may be very close to 3.

These results clearly follow the pattern predicted by [10, Theorems 4.60, 4.61]. For $k > \lambda \in (3,5)$ there exists a non-trivial steady state, and for $k < \lambda \in (3,5)$ the trivial steady state is stable, as expected.

The method proposed in this thesis being essentially domain independent provides a useful tool for investigating the properties of such "critical exponents" as the shape of the domain Ω changes.

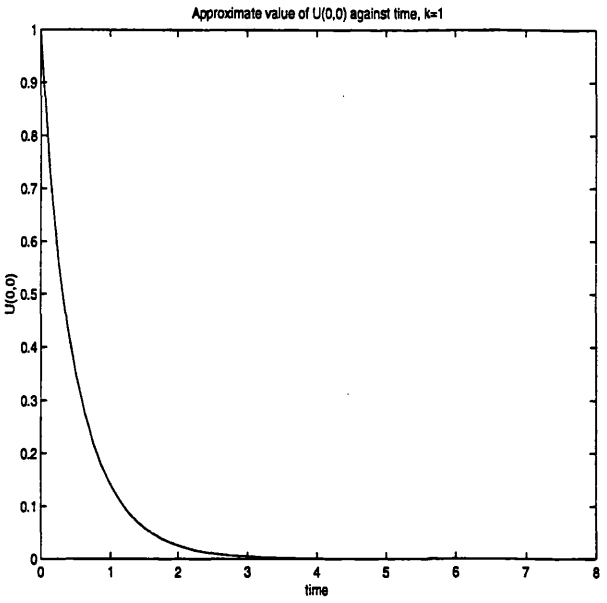

Figure 6-1: Convergence to trivial steady state, $k = 1$.

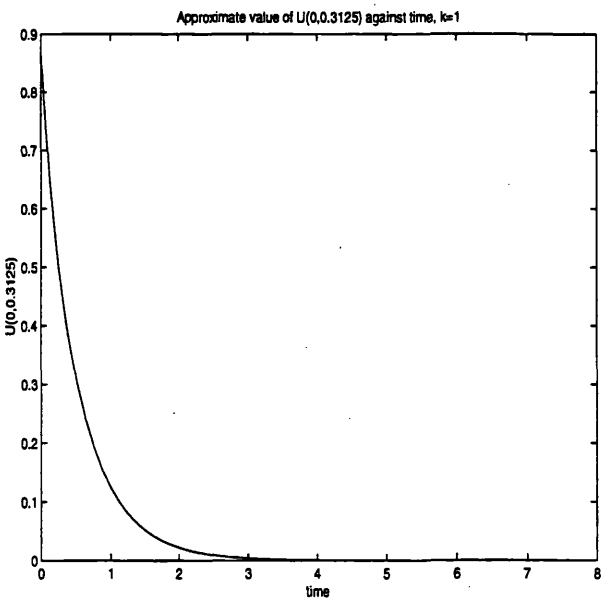

Figure 6-2: Convergence to trivial steady state, $k = 1$.

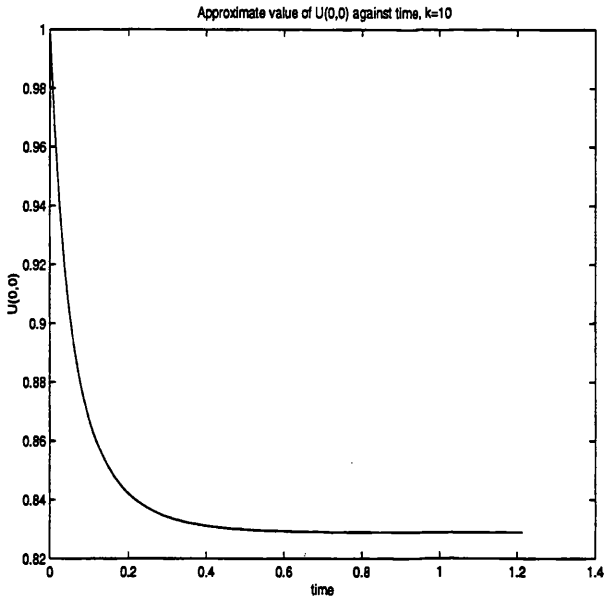

Figure 6-3: Convergence to non-trivial steady state, $k = 10$.

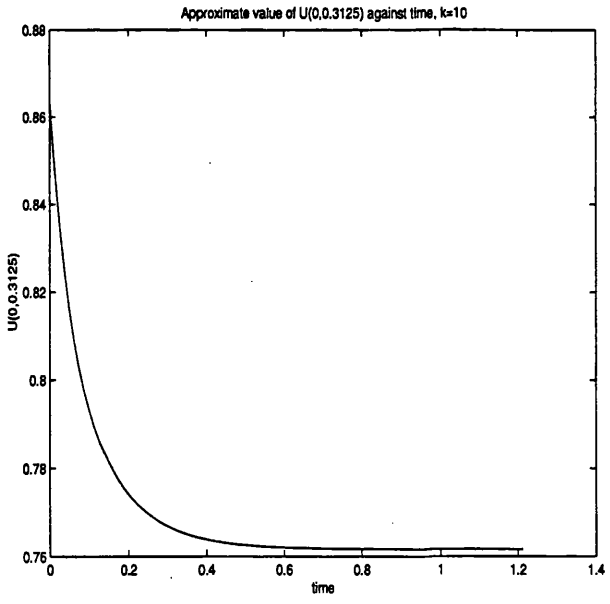

Figure 6-4: Convergence to non-trivial steady state, $k = 10$.

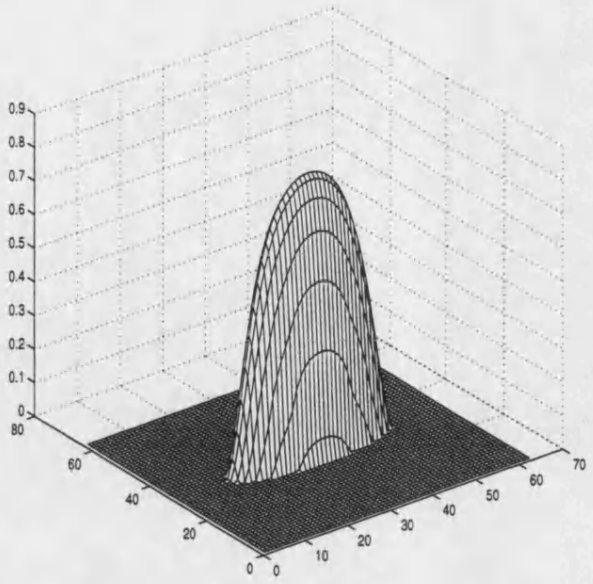

Figure 6-5: Non-trivial steady state, $k = 10$.

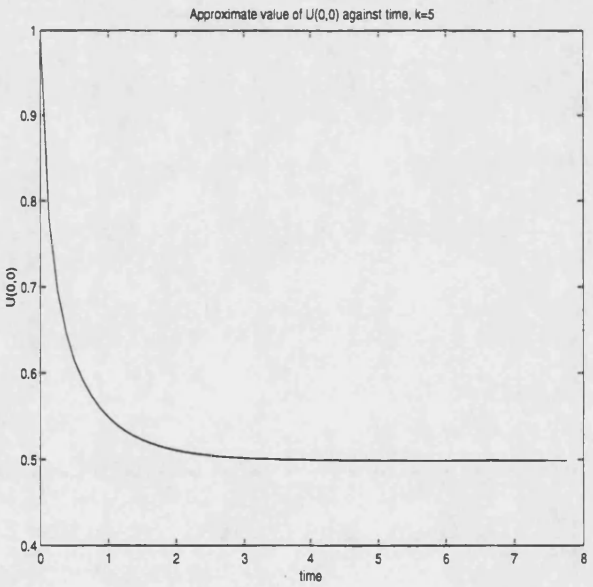

Figure 6-6: Convergence to non-trivial steady state, $k = 5$.

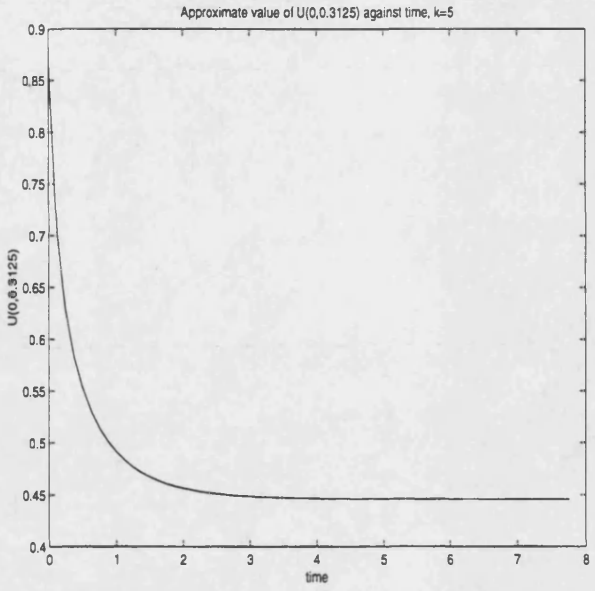

Figure 6-7: Convergence to non-trivial steady state, $k = 5$.

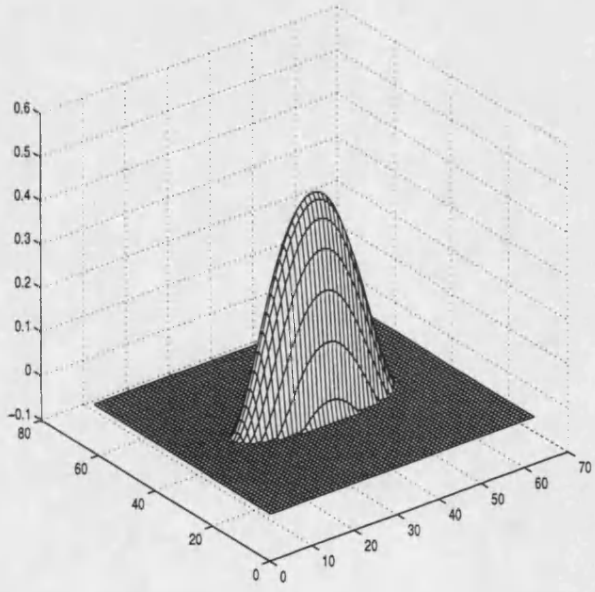

Figure 6-8: Non-trivial steady state, $k = 5$.

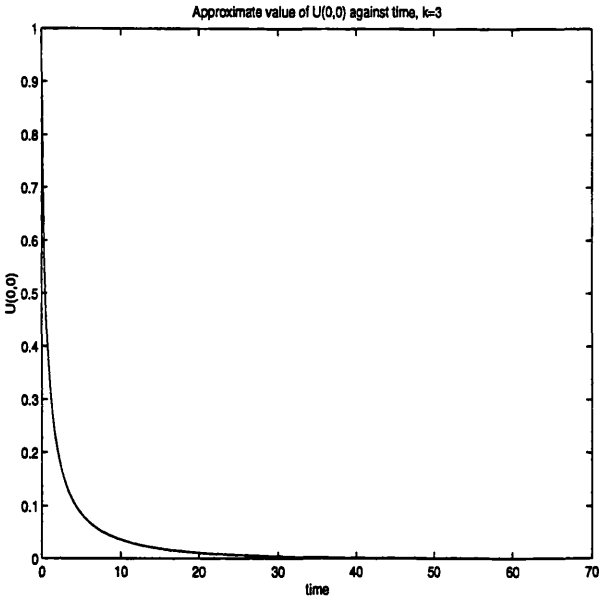

Figure 6-9: Convergence to trivial steady state, $k = 3$.

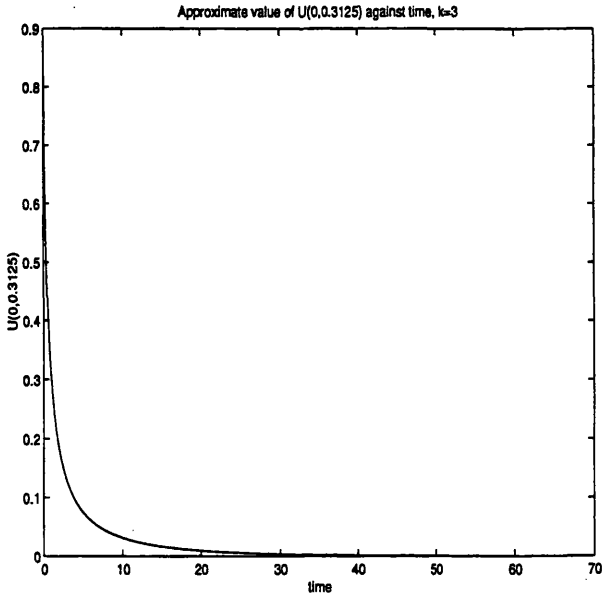

Figure 6-10: Convergence to trivial steady state, $k = 3$.

Chapter 7

Conclusions and future work

7.1 Conclusions

In this thesis we have developed a domain embedding boundary integral equation method for solving boundary value problems for the inhomogeneous Helmholtz-type equation

$$-\Delta u + \alpha^2 u = g, \quad \text{in } \Omega \subset \mathbb{R}^2, \tag{7.1.1}$$

where α is real, Ω is a bounded domain with a smooth boundary, and $g \in \mathcal{C}^{3,\gamma}(\bar{\Omega})$, $\gamma \in [0,1]$. Our method for solving (7.1.1) approximates u at $\mathcal{O}(n^2)$ points to $\mathcal{O}(n^{-2})$ accuracy and with a cost of $\mathcal{O}(n^2 \log n)$.

These asymptotic orders of cost and accuracy are the same as if we solved (7.1.1) using the finite element method on a uniform grid. However, although the methods are comparable in order, our method is essentially domain independent as it avoids altogether the explicit construction of a grid resolving Ω.

For the finite element method, the cost of generating a mesh with n points for a general domain Ω may be only $\mathcal{O}(n)$, but the constant may be large. Thus we hope that our method may prove to be a competitive alternative approach.

We have also considered in this thesis the extension of our method to the solution of certain parabolic PDEs, and although no attempt has been made to compare the performance of our method with existing methods for solving such problems, we would hope once again that it may be a competitive alternative.

7.2 Future work

Some areas for future research include:-

i). We would like to prove a fully discrete error bound for the method described in Chapter 5 for solving the heat equation.

ii). We would like to prove all of the results in Appendix C and Chapters 2 and 4 with error estimates independent of the parameter α (as we have already done in Chapter 3). In turn we would hope that this might lead to a full convergence analysis for the method described in Chapter 6 as it applies to the heat equation.

iii). We would like to perform further numerical experiments for the method of Chapter 3 with the parameter q_2 fixed, independently of α.

iv). We would like to investigate further the possible benefits of using the variable timestepping scheme described in Chapter 5.

v). In the future we hope to use our method to solve further challenging reaction diffusion problems, and to compare its performance for solving such problems directly with that of the finite element method.

Appendix A

Definitions of some function sets and spaces

Here we define some of the function sets and spaces which we use in this thesis. To begin with, for notational convenience we introduce the concept of a multi-index.

Definition A.0.1 *Let* $\mathbb{N}$ *denote the set of non-negative integers. Then*

$$m = (m_1, \ldots, m_n) \in \mathbb{N}^n$$

is a multi-index. *The order of* m *is defined by* $|m| := m_1 + \ldots + m_n$.

Now we define the space of continuous functions. Throughout this section, let Ω be an open bounded subset of $\mathbb{R}^n$ and let $\bar{\Omega}$ denote its closure.

Definition A.0.2 *If a function* f *is bounded on* Ω, *define*

$$\|f\|_{\infty,\Omega} := \sup_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x})|.$$

Now let $k \in \mathbb{N}$. *Then we denote by* $C^k(\Omega)$ *the set of all continuous functions* f *defined on* Ω *such that*

$$\mathcal{D}^m f := \frac{\partial^{|m|} f}{\partial x_1^{m_1} \ldots \partial x_n^{m_n}}$$

is continuous on Ω *for all* $m = (m_1, \ldots, m_n)$ *with* $|m| \leq k$. *By* $C^k(\bar{\Omega})$ *we denote the set of all functions* $f \in C^k(\Omega)$ *such that* $\mathcal{D}^m f$ *can be extended uniquely from*

Ω *to a continuous function on* $\bar{\Omega}$ *for all* $|m| \leq k$. *We equip* $C^k(\bar{\Omega})$ *with the norm:*

$$\|f\|_{k,\bar{\Omega}} := \max_{|m|\leq k} \|\mathcal{D}^m f\|_{\infty,\Omega}.$$

A sufficient condition for the existence of such an extension is that $\mathcal{D}^m f$ *is bounded and uniformly continuous on* Ω *(see for example [2, §1.26]). Note that* $\|f\|_{0,\bar{\Omega}} = \|f\|_{\infty,\Omega}$.

Next we define the space of Hölder continuous functions, as in [15, §2.2].

Definition A.0.3 *For* $0 < \gamma \leq 1$, *let* $C^{m,\gamma}(\bar{\Omega})$ *denote the space of functions* $\phi \in C^m(\bar{\Omega})$ *such that*

$$|\mathcal{D}^m\phi(\boldsymbol{x}) - \mathcal{D}^m\phi(\boldsymbol{y})| \leq C|\boldsymbol{x} - \boldsymbol{y}|^\gamma,$$

for some constant $C > 0$, *depending only on* ϕ. *Then* $C^{m,\gamma}(\bar{\Omega})$ *is equipped with the norm*

$$\|\phi\|_{m,\gamma,\bar{\Omega}} = \|\phi\|_{m,\bar{\Omega}} + \sup_{\substack{\boldsymbol{x},\boldsymbol{y}\in\bar{\Omega}\\ \boldsymbol{x}\neq\boldsymbol{y}}} \frac{|\mathcal{D}^m\phi(\boldsymbol{x}) - \mathcal{D}^m\phi(\boldsymbol{y})|}{|\boldsymbol{x} - \boldsymbol{y}|^\gamma}.$$

Remark: Clearly $C^{m,\gamma}(\bar{\Omega}) \subset C^m(\bar{\Omega}) \subset C^{m-1,\gamma}(\bar{\Omega})$, for all $m \geq 1$ and $\gamma \in (0, 1]$.

We now define spaces of integrable functions.

Definition A.0.4 *For* $p \geq 1$ *let* $L_p(\Omega)$ *denote the set of all Lebesgue measurable functions defined on* Ω *such that*

$$\int_\Omega |u(x)|^p dx < \infty.$$

$L_p(\Omega)$ *is then equipped with the norm*

$$\|u\|_{L_p(\Omega)} := \left(\int_\Omega |u(x)|^p dx\right)^{\frac{1}{p}}.$$

When $\Omega = [0, 2\pi]$ *we omit it from the notation.*

Finally we define Sobolev spaces. We only need these for 2π-periodic functions of one variable, so we can define them as in [49, Chapter 8].

Definition A.0.5 *For $s \geq 0$, let $H^s = H^s[0, 2\pi]$ denote the space of 2π-periodic locally Lebesgue integrable functions $v : \mathbb{R} \mapsto \mathbb{C}$ such that*

$$\|v\|_{H^s}^2 := \sum_{m=-\infty}^{\infty} (1 + m^2)^s |\hat{v}_m|^2 < \infty, \tag{A.0.1}$$

where

$$\hat{v}_m = \frac{1}{2\pi} \int_0^{2\pi} v(t) e^{-imt} dt, \quad m = 0, \pm 1, \pm 2, \ldots$$

are the Fourier coefficients of v. An equivalent norm exists for integers, and in particular if $v \in H^0$ then

$$\|v\|_{H^0} = \|v\|_{L_2},$$

and if $v \in H^1$ then the distributional derivative of v, v', satisfies $v' \in L_2$ and

$$\|v\|_{H^1}^2 = \|v\|_{L_2}^2 + \|v'\|_{L_2}^2.$$

Appendix B

Properties of modified Bessel functions

Throughout this thesis, we need to use various properties of the modified Bessel functions $K_\nu(z)$ (in particular $K_0(z)$ and $K_1(z)$), which are solutions of the differential equation

$$z^2\frac{d^2w}{dz^2} + z\frac{dw}{dz} - (z^2+\nu^2)w = 0.$$

All of the results we need can be found in either [1] or [57], but for ease of reference we summarise the most important ones here.

For real z, $K_\nu(z) \to 0$, as $|z| \to \infty$. Also $K_\nu(z)$ is real and positive when $z > 0$. The modified Bessel functions are related to Hankel functions $H_\nu^{(m)}$ via the formula

$$K_\nu(z) = \frac{\pi i}{2} e^{\frac{\nu\pi i}{2}} H_\nu^{(1)}(iz), \tag{B.0.1}$$

and so in particular

$$K_0(z) = \frac{\pi i}{2} H_0^{(1)}(iz). \tag{B.0.2}$$

When $|z|$ is small, there is the series representation [1, (9.6.11)] for $K_\nu(z)$,

which we state here.

$$K_\nu(z) = \frac{1}{2}\left(\frac{1}{2}z\right)^{-\nu}\sum_{k=0}^{\nu-1}\frac{(\nu-k-1)!}{k!}\left(-\frac{1}{4}z^2\right)^k + (-1)^{\nu+1}\log\left(\frac{1}{2}z\right)I_\nu(z)$$
$$+ (-1)^\nu\frac{1}{2}\left(\frac{1}{2}z\right)^\nu\sum_{k=0}^{\infty}a_{k,\nu}\frac{(\frac{1}{4}z^2)^k}{k!(\nu+k)!}, \qquad \text{(B.0.3)}$$

where

$$I_\nu(z) = \left(\frac{1}{2}z\right)^\nu\sum_{k=0}^{\infty}\frac{(\frac{1}{4}z^2)^k}{k!\Gamma(\nu+k+1)}.$$

From this we can deduce that as $z \to 0$,

$$K_0(z) \sim -\log(z) \text{ and } K_1(z) \sim \frac{1}{z},$$

where $\sim$ means "asymptotically equal". When z is real and large, there is an asymptotic expansion [1, (9.7.2)] for $K_\nu(z)$, from which we can deduce that, as $z \to \infty$,

$$K_0(z) \sim \frac{\pi}{\sqrt{2z}}e^{-z}\left\{1 - \frac{1}{8z}\right\}, \text{ and } K_1(z) \sim \frac{\pi}{\sqrt{2z}}e^{-z}\left\{1 + \frac{3}{8z}\right\}.$$

Finally, there are recurrence relations [1, (9.6.26)] for the derivatives of $K_\nu(z)$, from which we can deduce that

$$K_0'(z) = -K_1(z) \text{ and } K_1'(z) = -K_0(z) - \frac{1}{z}K_1(z). \qquad \text{(B.0.4)}$$

Appendix C

Continuity of the Volume Potential

Consider the volume potential

$$\mathcal{P}\psi(\boldsymbol{x}) := \int_\Omega \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y})d\boldsymbol{y}, \tag{C.0.1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with boundary Γ, and

$$\Phi_\alpha(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2\pi}K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|), \quad \boldsymbol{x} \neq \boldsymbol{y}, \tag{C.0.2}$$

is the fundamental solution of the two-dimensional Helmholtz equation, given by (2.1.5). In this section we show that under appropriate smoothness assumptions on ψ and Γ, $\mathcal{P}\psi$ is smooth enough for our needs, a result which is made more precise below and which is needed for the analysis in Chapter 2 (note that using the notation of Chapter 2, the domain integral U (2.1.6) is given by $U = \mathcal{P}g$). This result (on the smoothness of $\mathcal{P}g$) is by no means obvious, as in general an operator of the form (C.0.1), with a weakly singular kernel such as (C.0.2), may map smooth data ψ into a function $\mathcal{P}\psi$ with singularities on Γ. For example:-

Example C.0.1 *Consider the operator $\mathcal{K}$ defined by*

$$\mathcal{K}u(x) = \int_0^1 \log|x-y|u(y)dy, \quad x \in [0,1].$$

Now the function $u(x) = 1$, $x \in [0,1]$, *clearly satisfies* $u \in C^\infty[0,1]$, *but*

$$\begin{aligned} \mathcal{K}u(x) &= \int_0^1 \log|x-y|dy \\ &= x\log x + (1-x)\log(1-x) - 1, \end{aligned}$$

and so $\mathcal{K}u$ *is not infinitely smooth on* $[0,1]$.

The thing about the special integral $\mathcal{P}\psi$ defined in (C.0.1) is that $U := \mathcal{P}\psi$ furnishes a solution to the Helmholtz equation, ie.

$$(-\Delta + \alpha^2)U(\boldsymbol{x}) = \psi(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega.$$

So, if $\psi \in C^\infty(\bar{\Omega})$ and Γ and $U|_\Gamma$ are both smooth, then we would expect from the theory of regularity of elliptic PDEs (eg. [22, Theorem 6.19]) that U would be smooth on Ω. What is not so obvious is that $U|_\Gamma$ is a smooth function on Γ, and so a direct appeal to elliptic theory does not give us the result we require. The main purpose of this section is in fact to prove the smoothness of U using integral equation arguments. Our main result is:

Theorem C.0.2 *Suppose* $k \geq 1$ *is a given integer, and suppose* $\Gamma \in C^{k+1}$. *Let* $\gamma \in (0,1)$. *Then*

$$\mathcal{P} : C^{l,\gamma}(\bar{\Omega}) \mapsto C^{l+1,\gamma}(\bar{\Omega}), \text{ with } \|\mathcal{P}\psi\|_{l+1,\gamma,\bar{\Omega}} \leq C_{1,\gamma}\|\psi\|_{l,\gamma,\bar{\Omega}}, \quad l = 0,\ldots,k, \tag{C.0.3}$$

and

$$\mathcal{P} : C^{l,\gamma}(\bar{\Omega}) \mapsto C^{l+1,\gamma}(\mathbb{R}^2 \setminus \Omega), \text{with } \|\mathcal{P}\psi\|_{l+1,\gamma,\mathbb{R}^2\setminus\Omega} \leq C_{2,\gamma}\|\psi\|_{l,\gamma,\bar{\Omega}}, \quad l = 0,\ldots,k, \tag{C.0.4}$$

where $C_{1,\gamma}$, $C_{2,\gamma}$ *are constants independent of* ψ.

Proof: The proof of this result is complicated, and will take up the rest of this section.

Remark: We note that although results closely related to this appear in the literature (eg. Günter [33] has proved its analogue for the three-dimensional Laplace

operator, and Kangro [43] has proved a related result regarding the smoothness of the solution of a 2D integral equation with logarithmic kernel (see also [82, p.49–50])), we have not found this exact result anywhere, and so although the proof is somewhat lengthy, it is our belief that it is of interest. Note that we explicitly need the result for $k = 3$ in Chapter 2.

In our proof we follow the lines of the proof of Theorem 3 on p.108 of [33], for the three-dimensional Laplace equation. We also use results from [47] and [15] regarding the continuity of the single and double layer potentials defined by

$$\mathcal{V}\psi(\boldsymbol{x}) = \int_\Gamma \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\psi(\boldsymbol{y})d\boldsymbol{y} \tag{C.0.5}$$

and

$$\mathcal{W}\psi(\boldsymbol{x}) = \int_\Gamma \frac{\partial \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})}{\partial n(\boldsymbol{y})}\psi(\boldsymbol{y})d\boldsymbol{y} \tag{C.0.6}$$

respectively, where $\partial/\partial n(\boldsymbol{y})$ is the derivative with respect to the unit outward normal at $\boldsymbol{y} \in \Gamma$. Note that all the results we use from [47] and [15] are proved therein in the three-dimensional case, and so where appropriate we have worked through the proofs to demonstrate their applicability in two dimensions also. Note that as well as extending the result in [33] so that we can apply it to the volume potential arising from the two-dimensional Helmholtz equation, we have also simplified it somewhat, to make it less general, but more easily applicable to our needs.

The proof of Theorem C.0.2 is given at the end of this section. The main result we need in order to prove it is the following. We remind the reader that ∇ denotes the two-dimensional gradient operator. For a function of $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$ we use $\nabla_{\boldsymbol{x}}$ (respectively ∇_y) to denote the gradient with respect to x (respectively y).

Theorem C.0.3 *If $\Gamma \in \mathcal{C}^2$, and $\psi \in \mathcal{C}^1(\bar{\Omega})$, then we have the following formula for $\nabla\mathcal{P}\psi$:*

$$\nabla\mathcal{P}\psi(\boldsymbol{x}) = \mathcal{P}(\nabla\psi)(\boldsymbol{x}) - \mathcal{V}(\psi\boldsymbol{n})(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^2, \tag{C.0.7}$$

where $\boldsymbol{n} : \Gamma \mapsto \mathbb{R}^2$ represents the unit outward normal vector to Γ.

Proof: From (C.0.1) and using (B.0.4) we have

$$\begin{aligned}\nabla \mathcal{P}\psi(\boldsymbol{x}) &= \frac{1}{2\pi}\int_{\Omega} \nabla_x K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})d\boldsymbol{y} \\ &= \frac{\alpha}{2\pi}\int_{\Omega} \frac{(\boldsymbol{y}-\boldsymbol{x})}{|\boldsymbol{x}-\boldsymbol{y}|} K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})d\boldsymbol{y}. \qquad \text{(C.0.8)}\end{aligned}$$

First we prove (C.0.7) for $\boldsymbol{x} \in \Omega$. Let σ be a disk of radius δ around $\boldsymbol{x}$, with boundary $\partial\sigma$, as shown in Figure C-1 below. For small enough δ this disk lies completely inside Ω.

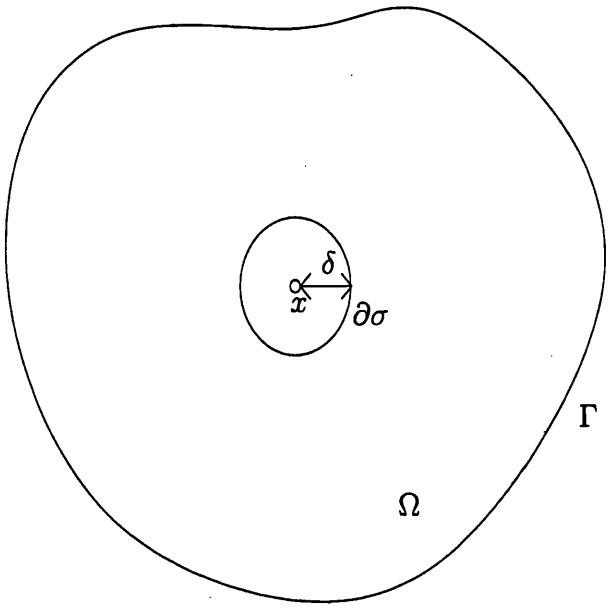

Figure C-1: Circle radius δ around $\boldsymbol{x} \in \Omega$.

We can then write (C.0.8) as

$$\begin{aligned}\nabla \mathcal{P}\psi(\boldsymbol{x}) &= \frac{\alpha}{2\pi}\int_{\sigma} \frac{(\boldsymbol{y}-\boldsymbol{x})}{|\boldsymbol{x}-\boldsymbol{y}|} K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})d\boldsymbol{y} \\ &\quad + \frac{\alpha}{2\pi}\int_{\Omega\setminus\sigma} \frac{(\boldsymbol{y}-\boldsymbol{x})}{|\boldsymbol{x}-\boldsymbol{y}|} K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})d\boldsymbol{y} \\ &=: \boldsymbol{A}+\boldsymbol{B}.\end{aligned}$$

First, we show that $|\boldsymbol{A}| \to 0$ as $\delta \to 0$. To do this note that it is clear that

$$\left| \frac{(\boldsymbol{y}-\boldsymbol{x})}{|\boldsymbol{x}-\boldsymbol{y}|} K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y}) \right| \leq K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\|\psi\|_{\infty,\Omega}.$$

If we then take polar coordinates about $\boldsymbol{x}$ we see that

$$\begin{aligned}
|\boldsymbol{A}| &\leq \frac{\alpha}{2\pi}\int_0^{2\pi}\int_0^{\delta} K_1(\alpha r) r dr d\theta \, \|\psi\|_{\infty,\Omega} \\
&= \frac{1}{\alpha}\int_0^{\alpha\delta} x K_1(x) dx \, \|\psi\|_{\infty,\Omega} \\
&= \frac{1}{\alpha}\left\{ [-xK_0(x)]_0^{\alpha\delta} + \int_0^{\alpha\delta} K_0(x) dx \right\} \|\psi\|_{\infty,\Omega} \quad \text{(integrating by parts)} \\
&= \frac{1}{\alpha}\{-\alpha\delta K_0(\alpha\delta) + \mathcal{O}(\delta\log(\delta))\} \, \|\psi\|_{\infty,\Omega} \to 0 \text{ as } \delta \to 0,
\end{aligned}$$

where in the last line we have used [1, (11.1.9)], which gives a series expansion for $\int_0^z K_0(x)dx$, and also the fact that $K_0(x) = \mathcal{O}(\log x)$ as $x \to 0$, see [1, (9.6.13)].

Now, to consider the limit of $\boldsymbol{B}$ as $\delta \to 0$, we write

$$\begin{aligned}
\boldsymbol{B} &= \frac{1}{2\pi}\int_{\Omega\setminus\sigma} \nabla_x K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y}) d\boldsymbol{y} \\
&= -\frac{1}{2\pi}\int_{\Omega\setminus\sigma} \nabla_y K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y}) d\boldsymbol{y} \\
&= -\frac{1}{2\pi}\int_{\Omega\setminus\sigma} \nabla_y \{K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})\} d\boldsymbol{y} \\
&\quad + \frac{1}{2\pi}\int_{\Omega\setminus\sigma} K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\nabla\psi(\boldsymbol{y}) d\boldsymbol{y} \qquad \text{(C.0.9)} \\
&=: \boldsymbol{C} + \boldsymbol{D}.
\end{aligned}$$

First we examine $\boldsymbol{D}$. We have

$$\begin{aligned}
\boldsymbol{D} &= \int_\Omega \frac{1}{2\pi} K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\nabla\psi(\boldsymbol{y}) d\boldsymbol{y} - \frac{1}{2\pi}\int_\sigma K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\nabla\psi(\boldsymbol{y}) d\boldsymbol{y} \\
&= \mathcal{P}(\nabla\psi)(\boldsymbol{x}) - \boldsymbol{E},
\end{aligned}$$

where

$$\boldsymbol{E} = \frac{1}{2\pi}\int_\sigma K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\nabla\psi(\boldsymbol{y}) d\boldsymbol{y}.$$

Taking polar coordinates about $\boldsymbol{x}$, we then have

$$\begin{aligned} |\boldsymbol{E}| &\leq \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\delta} K_0(\alpha r) r dr d\theta \, \|\nabla\psi\|_{\infty,\Omega} \\ &= \frac{1}{\alpha^2}\int_0^{\alpha\delta} x K_0(x) dx \, \|\nabla\psi\|_{\infty,\Omega}. \end{aligned}$$

As above, it is easily shown from [1, (11.3.27),(9.6.11)] that

$$\int_0^{\alpha\delta} x K_0(x) dx = 1 - \alpha\delta K_1(\alpha\delta) = \mathcal{O}(\delta^2 \log\delta) \text{ as } \delta \to 0,$$

and so $|\boldsymbol{E}| \to 0$ as $\delta \to 0$. Now it just remains to consider $\boldsymbol{C}$. Using the divergence theorem, we have

$$\begin{aligned} \boldsymbol{C} &= -\frac{1}{2\pi}\int_\Gamma K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})d\boldsymbol{y} - \frac{1}{2\pi}\int_{\partial\sigma} K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})\boldsymbol{n}_\sigma(\boldsymbol{y})d\boldsymbol{y} \\ &= -\mathcal{V}(\psi\boldsymbol{n})(\boldsymbol{x}) - \boldsymbol{F}, \end{aligned}$$

where

$$\boldsymbol{F} = \frac{1}{2\pi}\int_{\partial\sigma} K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})\boldsymbol{n}_\sigma(\boldsymbol{y})d\boldsymbol{y},$$

and where $\boldsymbol{n}_\sigma$ represents the unit outward normal vector to $\partial\sigma$. (Recall that $\boldsymbol{n}$ denotes the outward normal on Γ.) Using the fact that σ is the circle of radius δ, with centre $\boldsymbol{x}$, it can be easily shown that

$$|\boldsymbol{F}| \leq \delta K_0(\alpha\delta)\,\|\psi\|_{\infty,\Omega} \to 0 \text{ as } \delta \to 0,$$

using [1, (9.6.13)], and hence the result (C.0.7) follows, for $\boldsymbol{x} \in \Omega$.

To show that (C.0.7) holds for all $\boldsymbol{x} \in \bar{\Omega}$, we can consider it for $\boldsymbol{x} \in \Omega$, and let $\boldsymbol{x}$ tend to a point on Γ. Since $\nabla\psi \in \mathcal{C}^0(\bar{\Omega})$, it follows from Theorem C.0.10 below that $\mathcal{P}\nabla\psi \in \mathcal{C}^1(\bar{\Omega})$. Also, it is well known (see for example [15, Theorem 2.12] or Theorem C.0.7 below) that the single layer potential with continuous density is uniformly continuous throughout $\mathbb{R}^2$. Hence (C.0.7) holds for all $\boldsymbol{x} \in \bar{\Omega}$.

To show that (C.0.7) holds for $\boldsymbol{x} \in \mathbb{R}^2 \setminus \Omega$, note that we can write

$$\nabla \mathcal{P}\psi(\boldsymbol{x}) = -\frac{1}{2\pi}\int_\Omega \nabla_y\{K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\psi(\boldsymbol{y})\}d\boldsymbol{y} + \frac{1}{2\pi}\int_\Omega K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)\nabla\psi(\boldsymbol{y})d\boldsymbol{y},$$

as in (C.0.9), and the result follows by again using the divergence theorem to write the first integral on the right hand side as a boundary integral. □

Before we can use Theorem C.0.3, we need a theorem relating the continuity of $\mathcal{V}\psi$ to that of ψ. In the rest of this section we assume that $\Gamma \in \mathcal{C}^2$, and we let γ denote arc length parametrisation of Γ. Then, with $L = |\Gamma|$, $\gamma : [0, L] \mapsto \Gamma$ is L-periodic, $\gamma \in \mathcal{C}^2[0, L]$, and $|\gamma'(t)| = 1$, for all $t \in [0, L]$. Hence $\boldsymbol{\tau}(t) := \gamma'(t)$ is a unit tangent vector to Γ at the point $\boldsymbol{x} = \gamma(t)$. Since $\gamma'(t).\gamma'(t) = 1$ for all t it follows that $\gamma''(t).\gamma'(t) = 0$, and so $\boldsymbol{N}(t) := \gamma''(t)/|\gamma''(t)|$ is a unit normal vector to Γ (which may point inward or outward). Since $\boldsymbol{\tau}$ and $\boldsymbol{N}$ are an orthogonal basis in $\mathbb{R}^2$, we can then write, for $\boldsymbol{x} \in \Gamma$,

$$\begin{aligned}\nabla\psi(\boldsymbol{x}) &= (\nabla\psi.\boldsymbol{N})(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x}) + (\nabla\psi.\boldsymbol{\tau})(\boldsymbol{x})\boldsymbol{\tau}(\boldsymbol{x}) \\ &= \frac{\partial\psi(\boldsymbol{x})}{\partial N}\boldsymbol{N}(\boldsymbol{x}) + \frac{\partial\psi(\boldsymbol{x})}{\partial\tau}\boldsymbol{\tau}(\boldsymbol{x}). \qquad \text{(C.0.10)}\end{aligned}$$

We define the *curvature* of Γ by $\kappa(t) := |\gamma''(t)|$, and so we can then write

$$\gamma''(t) = \kappa(t)\boldsymbol{N}(t). \qquad \text{(C.0.11)}$$

We then have the following result.

Lemma C.0.4 *Let $\psi \in \mathcal{C}^1(\Gamma)$. Then*

$$\int_\Gamma (\nabla\psi.\boldsymbol{\tau})(\boldsymbol{y})\boldsymbol{\tau}(\boldsymbol{y})d\boldsymbol{y} = -\int_\Gamma \psi(\boldsymbol{y})\kappa(\boldsymbol{y})\boldsymbol{N}(\boldsymbol{y})d\boldsymbol{y}. \qquad \text{(C.0.12)}$$

Proof: To prove (C.0.12) we make the substitution $\boldsymbol{y} = \boldsymbol{\gamma}(t)$, and integrate by parts to get

$$\begin{aligned}\int_\Gamma (\nabla\psi.\boldsymbol{\tau})(\boldsymbol{y})\boldsymbol{\tau}(\boldsymbol{y})d\boldsymbol{y} &= \int_\Gamma \frac{\partial\psi(\boldsymbol{y})}{\partial\tau}\boldsymbol{\tau}(\boldsymbol{y})d\boldsymbol{y} \\ &= \int_0^L \frac{d\psi(\boldsymbol{\gamma}(t))}{dt}\boldsymbol{\gamma}'(t)dt \\ &= [\boldsymbol{\gamma}'(t)\psi(\boldsymbol{\gamma}(t))]_0^L - \int_0^L \psi(\boldsymbol{\gamma}(t))\boldsymbol{\gamma}''(t)dt,\end{aligned}$$

and the result follows using the periodicity of $\boldsymbol{\gamma}$ and (C.0.11). □

We can now prove the following two theorems giving formulae for the gradients of the single and double layer potentials (C.0.5) and (C.0.6).

Theorem C.0.5 *Let $\psi \in \mathcal{C}^1(\Gamma)$. Then for $\boldsymbol{x} \in \mathbb{R}^2 \setminus \Gamma$,*

$$\nabla\mathcal{V}\psi(\boldsymbol{x}) = \mathcal{V}((\nabla\psi.\boldsymbol{\tau})\boldsymbol{\tau} + \psi\kappa\boldsymbol{N})(\boldsymbol{x}) - \mathcal{W}(\psi\boldsymbol{N})(\boldsymbol{x}). \tag{C.0.13}$$

Proof: The proof is similar to that of [47, Theorem 3.2(i)], where an analogous result is proved in three dimensions. For $\boldsymbol{x} \notin \Gamma$,

$$\begin{aligned}\nabla\mathcal{V}\psi(\boldsymbol{x}) &= \int_\Gamma \nabla_x\{\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\}\psi(\boldsymbol{y})d\boldsymbol{y} \\ &= -\int_\Gamma \nabla_y\{\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\}\psi(\boldsymbol{y})d\boldsymbol{y} \\ &= -\int_\Gamma \left[\frac{\partial\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial\tau(\boldsymbol{y})}\boldsymbol{\tau}(\boldsymbol{y}) + \frac{\partial\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial N(\boldsymbol{y})}\boldsymbol{N}(\boldsymbol{y})\right]\psi(\boldsymbol{y})d\boldsymbol{y}, \quad \text{by (C.0.10)}, \\ &= -\int_\Gamma \left\{\frac{\partial(\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y}))}{\partial\tau(\boldsymbol{y})}\boldsymbol{\tau}(\boldsymbol{y}) - \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\frac{\partial\psi(\boldsymbol{y})}{\partial\tau}\boldsymbol{\tau}(\boldsymbol{y})\right. \\ &\qquad \left. + \frac{\partial\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial N(\boldsymbol{y})}\psi(\boldsymbol{y})\boldsymbol{N}(\boldsymbol{y})\right\}d\boldsymbol{y}, \qquad (C.0.14)\end{aligned}$$

using the product rule. Now applying Lemma C.0.4 (with $\psi(\boldsymbol{y})$ replaced by $\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y})$) to the first term of (C.0.14) the result follows. □

Theorem C.0.6 *Let $\psi \in \mathcal{C}^1(\Gamma)$. Then for $\boldsymbol{x} \in \mathbb{R}^2 \setminus \Gamma$,*

$$\nabla\mathcal{W}\psi(\boldsymbol{x}) = -\alpha^2\mathcal{V}(\psi\boldsymbol{n})(\boldsymbol{x}) - \nabla \wedge \mathcal{V}(\nabla\psi \wedge \boldsymbol{n})(\boldsymbol{x}), \tag{C.0.15}$$

recalling that $\boldsymbol{n}$ *is the unit* outward *normal vector to* Γ.

Proof: Again the proof is similar to that of [47, Theorem 3.2(ii)]. First, we define what we mean by curl and $\wedge$ in 2D. If $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{y} = (y_1, y_2)$ then

$$\boldsymbol{x} \wedge \boldsymbol{y} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ x_1 & x_2 & 0 \\ y_1 & y_2 & 0 \end{vmatrix},$$

and if $\boldsymbol{f} = (f_1, f_2)^T$ then

$$\text{curl}\boldsymbol{f} = \nabla \wedge \boldsymbol{f} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ \partial/\partial x & \partial/\partial y & 0 \\ f_1 & f_2 & 0 \end{vmatrix}.$$

Now note that for $\boldsymbol{x} \notin \Gamma$

$$\frac{\partial \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})}{\partial n(\boldsymbol{y})} = \{\nabla_y \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\}.\boldsymbol{n}(\boldsymbol{y}) = -\nabla_x.\{\Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})\}.$$

Hence (with ∇ denoting ∇_x and using the vector identity $\nabla\text{div} = \Delta + \text{curl curl}$), we have

$$\begin{aligned} \nabla \mathcal{W}\psi(\boldsymbol{x}) &= \nabla\left[-\nabla.\int_\Gamma \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})\, d\boldsymbol{y}\right] \\ &= -\Delta \int_\Gamma \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})\, d\boldsymbol{y} \\ &\quad - \text{curl}\left[\text{curl}\int_\Gamma \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})\, d\boldsymbol{y}\right]. \end{aligned} \tag{C.0.16}$$

To obtain (C.0.15) from (C.0.16), first consider the first term on the right hand side of (C.0.16). From Definition 2.1.1 it is clear that any function of the form $u(\boldsymbol{x}) = \int_\Gamma \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\phi(\boldsymbol{y})d\boldsymbol{y}$ satisfies $(-\Delta + \alpha^2)u(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \notin \Gamma$. Hence

$$-\Delta \int_\Gamma \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})d\boldsymbol{y} = -\alpha^2 \int_\Gamma \Phi_\alpha(\boldsymbol{x}, \boldsymbol{y})\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})d\boldsymbol{y}. \tag{C.0.17}$$

Now consider the second term on the right hand side of (C.0.16). It can be shown that

$$\begin{aligned}
\operatorname{curl} \int_\Gamma \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})\,d\boldsymbol{y} &= \int_\Gamma \nabla_x \wedge \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})\,d\boldsymbol{y} \\
&= \int_\Gamma \nabla_x \left\{\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\right\}\psi(\boldsymbol{y}) \wedge \boldsymbol{n}(\boldsymbol{y})\,d\boldsymbol{y} \\
&= -\int_\Gamma \nabla_y \left\{\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\right\}\psi(\boldsymbol{y}) \wedge \boldsymbol{n}(\boldsymbol{y})\,d\boldsymbol{y} \\
&= -\int_\Gamma \left[\frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial \tau(\boldsymbol{y})}\psi(\boldsymbol{y})\boldsymbol{\tau}(\boldsymbol{y})\right] \wedge \boldsymbol{n}(\boldsymbol{y})\,d\boldsymbol{y} \\
&\quad -\int_\Gamma \left[\frac{\partial \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})}{\partial N(\boldsymbol{y})}\psi(\boldsymbol{y})\boldsymbol{N}(\boldsymbol{y})\right] \wedge \boldsymbol{n}(\boldsymbol{y})\,d\boldsymbol{y},
\end{aligned}$$

using (C.0.10). The second term on the right hand side is clearly equal to zero, since $\boldsymbol{N} = \pm\boldsymbol{n}$, and so using the product rule on the first term we have

$$\begin{aligned}
\operatorname{curl} \int_\Gamma \Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y})\boldsymbol{n}(\boldsymbol{y})\,d\boldsymbol{y} &= -\int_\Gamma \left[\frac{\partial\{\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y})\}}{\partial \tau(\boldsymbol{y})}\boldsymbol{\tau}(\boldsymbol{y})\right] \wedge \boldsymbol{n}(\boldsymbol{y})\,d\boldsymbol{y} \\
&\quad +\int_\Gamma \left[\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\frac{\partial \psi(\boldsymbol{y})}{\partial \tau}\boldsymbol{\tau}(\boldsymbol{y})\right] \wedge \boldsymbol{n}(\boldsymbol{y})d\boldsymbol{y}.
\end{aligned} \tag{C.0.18}$$

Because $\boldsymbol{\tau}$ and $\boldsymbol{n}$ are orthogonal vectors in $\mathbb{R}^2$, the first term on the right hand side can be written as

$$-\int_\Gamma \frac{\partial\{\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y})\}}{\partial \tau(\boldsymbol{y})}(\boldsymbol{\tau} \wedge \boldsymbol{n})(\boldsymbol{y})\,d\boldsymbol{y} = -\int_\Gamma \frac{\partial\{\Phi_\alpha(\boldsymbol{x},\boldsymbol{y})\psi(\boldsymbol{y})\}}{\partial \tau(\boldsymbol{y})}(0,0,\pm 1)\,d\boldsymbol{y} = \boldsymbol{0},$$

using the fact that Γ is a closed contour. Taking the curl of (C.0.18) and combining with (C.0.17) in (C.0.16) the result (C.0.15) then follows. □

We are now nearly in a position to prove, by induction, our result Theorem C.0.9 on the continuity of $\mathcal{D}^l\mathcal{V}$ for $l > 1$, but first we need to establish the continuity properties of $\mathcal{V}\psi$ and $\mathcal{W}\psi$ on $\mathbb{R}^2$ when $\psi \in \mathcal{C}^{0,\gamma}(\Gamma)$. The following two theorems appear as Theorems 2.12 and 2.16 in [15], where they are proved for the three-dimensional case. The fundamental solution (C.0.2) of the two-dimensional Helmholtz equation is weakly singular at $\boldsymbol{x} = \boldsymbol{y}$, and decays exponentially when

$|\boldsymbol{x} - \boldsymbol{y}|$ is large, and this behaviour mirrors exactly that of the fundamental solution of the three-dimensional Helmholtz equation,

$$\Phi_{\alpha,3d}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{4\pi} \frac{e^{-\alpha|\boldsymbol{x}-\boldsymbol{y}|}}{|\boldsymbol{x} - \boldsymbol{y}|}.$$

Hence the proofs in two dimensions follow analogously, and so we do not prove these results here. Note that stronger results are possible, but Theorems C.0.7 and C.0.8 are sufficient for our needs.

Theorem C.0.7 *Suppose* $\Gamma \in \mathcal{C}^2$ *and* $\gamma \in (0,1)$. *Then*

$$\mathcal{V} : \mathcal{C}^0(\Gamma) \mapsto \mathcal{C}^{0,\gamma}(\mathbb{R}^2), \text{ with } \|\mathcal{V}\psi\|_{0,\gamma,\mathbb{R}^2} \leq C_\gamma \|\psi\|_{0,\Gamma},$$

where C_γ *is a constant depending only on* Γ *and* γ.

Theorem C.0.8 *Suppose* $\Gamma \in \mathcal{C}^2$ *and* $\gamma \in (0,1)$. *Then*

$$\mathcal{W} : \mathcal{C}^{0,\gamma}(\Gamma) \mapsto \mathcal{C}^{0,\gamma}(\bar{\Omega}), \text{ with } \|\mathcal{W}\psi\|_{0,\gamma,\bar{\Omega}} \leq C_{1,\gamma} \|\psi\|_{0,\gamma,\Gamma},$$

and

$$\mathcal{W} : \mathcal{C}^{0,\gamma}(\Gamma) \mapsto \mathcal{C}^{0,\gamma}(\mathbb{R}^2 \setminus \Omega), \text{ with } \|\mathcal{W}\psi\|_{0,\gamma,\mathbb{R}^2\setminus\Omega} \leq C_{2,\gamma} \|\psi\|_{0,\gamma,\Gamma},$$

where $C_{1,\gamma}$ *and* $C_{2,\gamma}$ *are constants depending only on* Γ *and* γ.

We can now prove the main result regarding the continuity of the derivatives of the single layer potential.

Theorem C.0.9 *Suppose* $k \geq 1$ *is a given integer, and suppose* $\Gamma \in \mathcal{C}^{k+1}$. *Let* $\gamma \in (0,1)$. *Then*

$$\mathcal{V} : \mathcal{C}^{l,\gamma}(\Gamma) \mapsto \mathcal{C}^{l,\gamma}(\bar{\Omega}), \text{ with } \|\mathcal{V}\psi\|_{l,\gamma,\bar{\Omega}} \leq C_{1,\gamma} \|\psi\|_{l,\gamma,\Gamma}, \quad l = 0, \ldots, k, \tag{C.0.19}$$

and

$$\mathcal{V} : \mathcal{C}^{l,\gamma}(\Gamma) \mapsto \mathcal{C}^{l,\gamma}(\mathbb{R}^2 \setminus \Omega), \text{ with } \|\mathcal{V}\psi\|_{l,\gamma,\mathbb{R}^2\setminus\Omega} \leq C_{2,\gamma} \|\psi\|_{l,\gamma,\Gamma}, \quad l = 0, \ldots, k, \tag{C.0.20}$$

where $C_{1,\gamma}$, $C_{2,\gamma}$ are constants independent of ψ.

Proof: We shall prove the result (C.0.19) using induction on l. Because the proof of (C.0.20) is similar, we shall omit it.

From Theorem C.0.7 it is clear that the result is true for $l = 0$. Assume the result is true for some $l \leq k-1$, and then suppose that $\psi \in \mathcal{C}^{l+1,\gamma}(\Gamma)$, $\gamma \in (0,1)$. Using the smoothness of the boundary we have that $\psi \boldsymbol{n} \in \mathcal{C}^{l+1,\gamma}(\Gamma)$ and $\nabla\psi \wedge \boldsymbol{n} \in \mathcal{C}^{l,\gamma}(\Gamma)$. Hence by the inductive hypothesis $\mathcal{V}(\psi\boldsymbol{n}) \in \mathcal{C}^{l,\gamma}(\bar{\Omega})$ and

$$\|\mathcal{V}(\psi\boldsymbol{n})\|_{l-1,\gamma,\bar{\Omega}} \leq \|\mathcal{V}(\psi\boldsymbol{n})\|_{l,\gamma,\bar{\Omega}} \leq C_{3,\gamma}\|\psi\|_{l,\gamma,\Gamma},$$

where the constant $C_{3,\gamma}$ is independent of ψ. Also, $\mathcal{V}(\nabla\psi \wedge \boldsymbol{n}) \in \mathcal{C}^{l,\gamma}(\bar{\Omega})$, and so $\nabla \wedge \mathcal{V}(\nabla\psi \wedge \boldsymbol{n}) \in \mathcal{C}^{l-1,\gamma}(\bar{\Omega})$ and

$$\|\nabla \wedge \mathcal{V}(\nabla\psi \wedge \boldsymbol{n})\|_{l-1,\gamma,\bar{\Omega}} = \|\mathcal{V}(\nabla\psi \wedge \boldsymbol{n})\|_{l,\gamma,\bar{\Omega}} \leq C_{4,\gamma}\|\nabla\psi\|_{l,\gamma,\Gamma} = C_{4,\gamma}\|\psi\|_{l+1,\gamma,\Gamma},$$

where the constant $C_{4,\gamma}$ is also independent of ψ. Using Theorem C.0.6 we see that $\nabla\mathcal{W}\psi \in \mathcal{C}^{l-1,\gamma}(\bar{\Omega})$, and hence

$$\mathcal{W}\psi \in \mathcal{C}^{l}(\bar{\Omega}), \tag{C.0.21}$$

and

$$\|\mathcal{W}\psi\|_{l,\gamma,\Omega} \leq C_{\gamma}\|\psi\|_{l+1,\gamma,\Gamma}, \tag{C.0.22}$$

where the constant C_{γ} is independent of ψ.

We can now use these results in conjunction with Theorem C.0.5 to get the result. Since $\psi \in \mathcal{C}^{l+1,\gamma}(\Gamma)$ and $\kappa\boldsymbol{N} \in \mathcal{C}^{k-1}(\Gamma)$, with $l \leq k-1$, it follows that $((\nabla\psi.\boldsymbol{\tau})\boldsymbol{\tau} + \psi\kappa\boldsymbol{N}) \in \mathcal{C}^{l,\gamma}(\Gamma)$, and so under the inductive hypothesis we have $\mathcal{V}((\nabla\psi.\boldsymbol{\tau})\boldsymbol{\tau} + \psi\kappa\boldsymbol{N}) \in \mathcal{C}^{l,\gamma}(\bar{\Omega})$, and

$$\|\mathcal{V}((\nabla\psi.\boldsymbol{\tau})\boldsymbol{\tau} + \psi\kappa\boldsymbol{N})\|_{l,\gamma,\bar{\Omega}} \leq C_{\gamma}\|\psi\|_{l+1,\gamma,\Gamma}, \tag{C.0.23}$$

where the constant C_γ is independent of ψ. Using this, (C.0.21) and Theorem C.0.5 we then have that $\nabla \mathcal{V}\psi \in \mathcal{C}^{l,\gamma}(\bar{\Omega})$, and so

$$\mathcal{V}\psi \in \mathcal{C}^{l+1,\gamma}(\bar{\Omega}), \tag{C.0.24}$$

and from (C.0.23), (C.0.22) and (C.0.13) we see that

$$\|\mathcal{V}\psi\|_{l+1,\gamma,\Omega} \leq C_{1,\gamma}\|\psi\|_{l+1,\gamma,\Gamma}. \tag{C.0.25}$$

From (C.0.24) and (C.0.25), it is then clear that the result (C.0.19) holds for $l+1$ under the assumption that it is true for l, and so by induction the result is true for all $l = 0, \ldots, k$. □

We are now nearly in a position to prove Theorem C.0.2. Again the proof is by induction and the next theorem proves the first step in the induction.

Theorem C.0.10 *Suppose $\Gamma \in \mathcal{C}^2$ and $\gamma \in (0,1)$. Then*

$$\mathcal{P} : \mathcal{C}^0(\bar{\Omega}) \mapsto \mathcal{C}^{1,\gamma}(\bar{\Omega}), \text{ with } \|\mathcal{P}\psi\|_{1,\gamma,\bar{\Omega}} \leq C_{1,\gamma}\|\psi\|_{0,\bar{\Omega}}, \tag{C.0.26}$$

and

$$\mathcal{P} : \mathcal{C}^0(\bar{\Omega}) \mapsto \mathcal{C}^{1,\gamma}(\mathbb{R}^2 \setminus \Omega), \text{ with } \|\mathcal{P}\psi\|_{1,\gamma,\mathbb{R}^2\setminus\Omega} \leq C_{2,\gamma}\|\psi\|_{0,\bar{\Omega}}, \tag{C.0.27}$$

where $C_{1,\gamma}$ and $C_{2,\gamma}$ are constants independent of ψ.

Proof: We prove (C.0.26) only, the proof of (C.0.27) is similar. Let $\psi \in \mathcal{C}^0(\bar{\Omega})$. To show $\mathcal{P}\psi \in \mathcal{C}^{1,\gamma}(\bar{\Omega})$, we need to show that $\nabla\mathcal{P}\psi \in \mathcal{C}^{0,\gamma}(\bar{\Omega})$, where $\nabla\mathcal{P}\psi$ is given by (C.0.8). To do this we will need to use [44, Theorem 4, p.363], which states that if the kernel

$$R(\boldsymbol{x},\boldsymbol{y}) = \frac{(\boldsymbol{y}-\boldsymbol{x})}{|\boldsymbol{x}-\boldsymbol{y}|}K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)$$

of (C.0.8) satisfies the two conditions

$$\int_\Omega |\nabla_x R(\boldsymbol{x},\boldsymbol{y})|\,|\boldsymbol{x}-\boldsymbol{y}|^{1-\gamma}d\boldsymbol{y} \leq C_1 \tag{C.0.28}$$

and

$$\int_\Omega |R(\boldsymbol{x},\boldsymbol{y})|\,|\boldsymbol{x}-\boldsymbol{y}|^{-\gamma}d\boldsymbol{y} \le C_2, \tag{C.0.29}$$

then $\nabla\mathcal{P} : L^\infty(\bar\Omega) \mapsto \mathcal{C}^{0,\gamma}(\bar\Omega)$, and

$$\|\nabla\mathcal{P}\psi\|_{0,\gamma,\bar\Omega} \le [C_1 + (2^\gamma + 3^\gamma)C_2]\,\|\psi\|_{0,\bar\Omega}.$$

So, we just need to establish the bounds (C.0.28) and (C.0.29), and then the result will follow. Using the formula for the derivative of $K_1(z)$ (see Appendix B) we have

$$\begin{aligned}
&\int_\Omega |\nabla_x R(\boldsymbol{x},\boldsymbol{y})|\,|\boldsymbol{x}-\boldsymbol{y}|^{1-\gamma}d\boldsymbol{y} = \\
&\qquad \int_\Omega \left| \frac{\alpha(\boldsymbol{y}-\boldsymbol{x})^2}{|\boldsymbol{x}-\boldsymbol{y}|^2} K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|) \right. \\
&\qquad\qquad \left. + \left\{ \frac{2(\boldsymbol{x}-\boldsymbol{y})^2 - |\boldsymbol{x}-\boldsymbol{y}|^2}{|\boldsymbol{x}-\boldsymbol{y}|^3} \right\} K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|) \right| |\boldsymbol{x}-\boldsymbol{y}|^{1-\gamma}d\boldsymbol{y} \\
&\le \int_\Omega \alpha|\boldsymbol{x}-\boldsymbol{y}|^{1-\gamma}K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)d\boldsymbol{y} + \int_\Omega |\boldsymbol{x}-\boldsymbol{y}|^{-\gamma}K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)d\boldsymbol{y} \\
&=: A + B.
\end{aligned}$$

Now, define $D(\boldsymbol{x})$ to be the circle of radius d and centre $\boldsymbol{x} \in \bar\Omega$, where d is chosen in such a way that $\Omega \subset D(\boldsymbol{x})$. Since Ω is a bounded domain, such a circle can certainly be defined for each $\boldsymbol{x} \in \bar\Omega$ by choosing d sufficiently large. Using the series expansions for $K_0(z)$ and $K_1(z)$ (see Appendix B), it can be shown that there exist constants C_3 and C_4 such that for any $z \in [0,d]$, $d > 0$,

$$K_0(z) \le C_3 K_1(z) \le C_4 \frac{1}{z}. \tag{C.0.30}$$

Stronger bounds also hold, but (C.0.30) is sufficient for our needs. Using it we get

$$\begin{aligned} |A| &\leq \int_D \alpha|\boldsymbol{x}-\boldsymbol{y}|^{1-\gamma}K_0(\alpha|\boldsymbol{x}-\boldsymbol{y}|)d\boldsymbol{y} \\ &= \int_0^d \int_0^{2\pi} \alpha r^{2-\gamma}K_0(\alpha r)dr\,d\theta \\ &\leq C\int_0^{\alpha d} t^{1-\gamma}dt, \end{aligned}$$

for some constant C. This is bounded for all $\gamma \in (0,1)$. Similarly, we have

$$\begin{aligned} |B| &\leq \int_D |\boldsymbol{x}-\boldsymbol{y}|^{-\gamma}K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)d\boldsymbol{y} \\ &= \int_0^d \int_0^{2\pi} r^{1-\gamma}K_1(\alpha r)dr\,d\theta \\ &\leq C\int_0^{\alpha d} t^{-\gamma}dt, \end{aligned}$$

which again is bounded for all $\gamma \in (0,1)$.

So, this shows that (C.0.28) is satisfied. To establish the bound (C.0.29), note that

$$\int_\Omega |R(\boldsymbol{x},\boldsymbol{y})|\,|\boldsymbol{x}-\boldsymbol{y}|^{-\gamma}d\boldsymbol{y} \leq \int_\Omega |\boldsymbol{x}-\boldsymbol{y}|^{-\gamma}K_1(\alpha|\boldsymbol{x}-\boldsymbol{y}|)d\boldsymbol{y},$$

and then the result follows using exactly the same argument as above. This completes the proof. □

Proof of Theorem C.0.2: We now complete the proof of Theorem C.0.2, using induction. We prove (C.0.3) only, the proof of (C.0.4) is analogous.

From Theorem C.0.10 the result is true for $l = 0$. Assume it is true for some $l \leq k-1$, and then let $\psi \in \mathcal{C}^{l+1,\gamma}(\bar{\Omega})$. Then $\nabla\psi \in \mathcal{C}^{l,\gamma}(\bar{\Omega})$, and so under the inductive hypothesis $\mathcal{P}(\nabla\psi) \in \mathcal{C}^{l+1,\gamma}(\bar{\Omega})$. We also have $\psi\boldsymbol{n} \in \mathcal{C}^{l+1,\gamma}(\bar{\Omega})$ by the assumed smoothness of Γ, and so by Theorem C.0.9 we have $\mathcal{V}(\psi\boldsymbol{n}) \in \mathcal{C}^{l+1,\gamma}(\bar{\Omega})$. So, using Theorem C.0.3 we have that $\nabla\mathcal{P}\psi \in \mathcal{C}^{l+1,\gamma}(\bar{\Omega})$, from which it follows that $\mathcal{P}\psi \in \mathcal{C}^{l+2,\gamma}(\bar{\Omega})$. In addition, using the inductive hypothesis and Theorem C.0.9 we have that there exists a constant C_γ independent of ψ such

that

$$\|\nabla \mathcal{P}\psi\|_{l+1,\gamma,\tilde{\Omega}} \leq C_\gamma \|\psi\|_{l+1,\gamma,\tilde{\Omega}},$$

and hence

$$\|\mathcal{P}\psi\|_{l+2,\gamma,\tilde{\Omega}} \leq C_\gamma \|\psi\|_{l+1,\gamma,\tilde{\Omega}},$$

as required. So, we have shown that the result is true for $l = 0$, and true for $l + 1$ under the assumption that it is true for l, and thus we have shown that Theorem C.0.2 is true for all $l = 0, \ldots, k$. This concludes the proof. □

Bibliography

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions.* Dover, New York, 1972.

[2] R.A. Adams. *Sobolev Spaces.* Academic Press, New York, 1975.

[3] D.N. Arnold and P.J. Noon. Coercivity of the single layer heat potential. *J. Comput. Math.*, 7(2):100–104, 1989.

[4] K. Atkinson. *A Survey of Numerical Methods for the Solution of Fredholm Integral Equations of the Second Kind.* SIAM, Philadelphia, 1976.

[5] K.E. Atkinson. The numerical evaluation of particular solutions for Poisson's equation. *IMA Journal of Numerical Analysis*, 5:319–338, 1985.

[6] K.E. Atkinson. A discrete Galerkin method for first kind integral equations with a logarithmic kernel. *Journal of Integral Equations and Applications*, 1(3):343–363, 1988.

[7] R. Beetson and L. Greengard. A short course on fast multipole methods. In M. Ainsworth, J. Levesley, W.A. Light, and M. Marletta, editors, *Wavelets, Multilevel Methods and Elliptic PDEs*, Numerical Mathematics and Scientific Computation. Oxford University Press, 1996.

[8] S. Brenner and L. Scott. *The Mathematical Theory of Finite Element Methods.* Springer-Verlag, 1994.

[9] William L. Briggs and Van Emden Henson. *The DFT: an owner's manual for the discrete Fourier transform.* SIAM, Philadelphia, 1995.

[10] N.F. Britton. *Reaction-Diffusion Equations and Their Applications to Biology.* Academic Press, 1986.

[11] S.N. Chandler-Wilde, J. Levesley, and D.M. Hough. Evaluation of a boundary integral representation for the conformal mapping of the unit disk onto a simply-connected domain. *Advances in Computational Mathematics*, 3:115–135, 1995.

[12] R. Chapko and R. Kress. Rothe's method for the heat equation and boundary integral equations. *Journal Integral Equations and Appl.*, 9:47–69, 1997.

[13] C.S. Chen and M.A. Golberg. A domain embedding method and the quasi-Monte-Carlo method for Poisson's equation. In *Boundary Element XVII*, pages 115–122. Computational Mechanics Publications, 1995.

[14] G. Chen and J. Zhou. *Boundary Element Methods.* Academic Press, London, 1992.

[15] D. Colton and R. Kress. *Integral Equation Methods in Scattering Theory.* Wiley, New York, 1983.

[16] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory.* Springer-Verlag, 1992.

[17] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19:297–301, 1965.

[18] M. Costabel. Boundary integral operators for the heat equation. *Integral Equations and Operator Theory*, 13(4):498–552, 1990.

[19] A. Friedman. *Partial Differential Equations of Parabolic Type.* Prentice-Hall, Englewood Cliffs, N.J., 1964.

[20] A. Friedman. *Partial Differential Equations.* Krieger, Malabar, Fl., 1983.

[21] W. Gerdes. Die lösung des anfangs-randwertproblems für die wärmeleitungsgleichung im $\mathbb{R}^3$ mit einer integralgleichungsmethode nach dem Rothe-verfahren. *Computing*, 19:251–268, 1978.

[22] D. Gilbarg and N.S. Trudinger. *Elliptic Partial Differential Equations of Second Order.* Springer-Verlag, Berlin, 1983.

[23] G. Steven Gipson. Use of the residue theorem in locating points within an arbitrary multiply-connected region. *Advances in Engineering Software and Workstations*, 8(2):73–80, 1986.

[24] M.A. Golberg. The method of fundamental solutions for Poisson's equation - II. *Engineering Analysis with Boundary Elements*, 16(3):205–213, 1995.

[25] M.A. Golberg. The numerical evaluation of particular solutions in the BEM - a review. *Boundary Element Communications*, 6(3):99–106, 1995.

[26] M.A. Golberg and C.S. Chen. On a method of Atkinson for evaluating domain integrals in the boundary element method. *Applied Mathematics and Computation*, 60:125–138, 1994.

[27] M.A. Golberg and C.S. Chen. The theory of radial basis functions applied to the BEM for inhomogeneous partial differential equations. *Boundary Elements Communications*, 5(2):57–61, 1994.

[28] J. Gold. Improved solutions to Laplace's equation by the boundary integral method. M.Phil. thesis, University of Bath, 1991.

[29] J. Gold and I.G. Graham. Towards automation of boundary integral methods for Laplace's equation. In J.R. Whiteman, editor, *The Mathematics of Finite Elements and Applications VII*, pages 349–359. Academic Press, 1991.

[30] I.G. Graham and K.E. Atkinson. On the Sloan iteration applied to integral equations of the first kind. *IMA Journal of Numerical Analysis*, 13:29–41, 1993.

[31] I.G. Graham and G.A. Chandler. High-order methods for linear functionals of solutions of second kind integral equations. *SIAM J. Numer. Anal.*, 25(5):1118–1137, 1988.

[32] I.G. Graham and W.R. Mendes. Nyström-product integration for Wiener-Hopf equations with applications to radiative transfer. *IMA Journal of Numerical Analysis*, 9:261–284, 1989.

[33] N.M. Günter. *Potential Theory and Its Applications to Basic Problems of Mathematical Physics.* Ungar, New York, 1967.

[34] W. Hackbusch. *Elliptic Differential Equations: Theory and Numerical Treatment.* Springer-Verlag, Berlin, 1992.

[35] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations. I. Nonstiff Problems.* Springer-Verlag, 1987.

[36] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems.* Springer-Verlag, 1991.

[37] Martti Hamina and Jukka Saranen. On the spline collocation method for the single-layer heat operator equation. *Mathematics of Computation*, 62(205):41–64, 1994.

[38] Olaf Hansen. On a boundary integral method for the solution of the heat equation in unbounded domains with a nonsmooth boundary. Preprint, July 1999.

[39] G.C. Hsiao and J. Saranen. Boundary integral solution of the two-dimensional heat equation. *Mathematical Methods in the Applied Sciences*, 16:87–114, 1993.

[40] Yuusuke Iso and Kazuei Onishi. On the stability of the boundary element collocation method applied to the linear heat equation. *Journal of Computational and Applied Mathematics*, 38:201–209, 1991.

[41] Fritz John. The fundamental solution of linear elliptic differential equations with analytic coefficients. In Jurgen Moser, editor, *Fritz John : collected papers*, Contemporary Mathematicians, pages 163–194. Birkhäuser, Boston, 1985.

[42] Lee W. Johnson and R. Dean Riess. *Numerical Analysis.* Addison-Wesley, second edition, 1982.

[43] U Kangro. The smoothness of the solution to a two-dimensional integral equation with logarithmic kernel. *Zeitschrift für Analysis und ihre Anwendungen*, 12:305–318, 1993.

[44] L.V. Kantorovich and G.P. Akilov. *Functional Analysis in Normed Spaces.* Pergamon Press, 1964.

[45] J. Kačur and R. Van Keer. On the numerical solution of a class of nonlinear parabolic problems with Volterra operators by a Rothe-Galerkin finite element method. *IMA Journal of Numerical Analysis*, 17:239–269, 1997.

[46] S. Kesavan and A.S. Vasudevamurthy. On some boundary element methods for the heat equation. *Numer. Math.*, 46:101–120, 1985.

[47] A. Kirsch. Surface gradients and continuity properties for some integral operators in classical scattering theory. *Mathematical Methods in the Applied Sciences*, 11:789–804, 1989.

[48] N. Kolkovska. Computation of the simple and double layer logarithmic potentials. Preprint, 1997.

[49] R. Kress. *Linear Integral Equations.* Springer-Verlag, 1989.

[50] R. Kress. A Nyström method for boundary integral equations in domains with corners. *Numer. Math.*, 58:145–161, 1990.

[51] R. Kress and I.H. Sloan. On the numerical solution of a logarithmic integral equation of the first kind for the Helmholtz equation. *Numer. Math.*, 66:199–214, 1993.

[52] S. Langdon. A boundary integral equation method for the heat equation. In A. Struthers and B. Bertram, editors, *Proceedings of the 5th international conference on integral methods in science and engineering (IMSE98), held at Michigan Technological University.* CRC Press, August 1998.

[53] S. Langdon and I.G. Graham. Boundary integral methods for singularly perturbed boundary value problems. Mathematics Preprint 98/19, University of Bath, September 1998. (submitted for publication).

[54] C. Lubich. Convolution quadrature and discretized operational calculus. i. *Numer. Math.*, 52:129–145, 1988.

[55] C. Lubich and A. Ostermann. Linearly implicit time discretization of nonlinear parabolic equations. *IMA Journal of Numerical Analysis*, 15:555–583, 1995.

[56] C. Lubich and R. Schneider. Time discretization of parabolic boundary integral equations. *Numer. Math.*, 63(4):455–481, 1992.

[57] Yudell L. Luke. *Integrals of Bessel Functions.* Mcgraw-Hill, 1962.

[58] A. Mayo. The fast solution of Poisson's and the biharmonic equations on irregular regions. *SIAM J. Numer. Anal.*, 21(2):285–299, 1984.

[59] A. Mayo. Fast high order accurate solution of Laplace's equation on irregular regions. *SIAM J. Sci Stat. Comput.*, 6(1):144–157, 1985.

[60] A. Mayo. The rapid evaluation of volume integrals of potential theory on general regions. *Journal of Computational Physics*, 100:236–245, 1992.

[61] A. Mayo and A. Greenbaum. Fast parallel iterative solution of Poisson's and the biharmonic equations on irregular regions. Technical Report 540, New York University, 1991.

[62] A. McKenney, L. Greengard, and A. Mayo. A fast Poisson solver for complex geometries. *Journal of Computational Physics*, 118(2):348–355, 1995.

[63] W. Mendes. *The Numerical Solution of Wiener-Hopf Integral Equations.* PhD thesis, University of Bath, 1988.

[64] J. Murray. *Mathematical Biology.* Springer-Verlag, 1989.

[65] F.W.J. Olver. *Asymptotics and Special Functions.* Academic Press, 1974.

[66] P.W. Partridge, C.A. Brebbia, and L.C. Wrobel. *The Dual Reciprocity Boundary Element Method.* Computational Mechanics Publications, 1992.

[67] M. Pickering. *An Introduction to Fast Fourier Transform Methods for PDEs with Applications.* Research Studies Press, 1986.

[68] M.J.D. Powell. The theory of radial basis function approximation in 1990. In Will Light, editor, *Advances in Numerical Analysis*, volume II. Oxford Science Publications, Oxford, 1992.

[69] M.J.D. Powell. Recent research at Cambridge on radial basis functions. Technical Report NA05, DAMTP, Cambridge, 1998.

[70] H.A. Priestley. *Introduction to Complex Analysis.* Oxford University Press, 1990.

[71] S. Prössdorf. *Some Classes of Singular Equations.* North-Holland Publishing Company, 1978.

[72] S. Prössdorf and B. Silbermann. *Numerical Analysis for Integral and Related Operator Equations.* Birkhäuser, 1991.

[73] J. Pryce. *Basic Methods of Linear Functional Analysis.* Hutchinson University Library, 1973.

[74] P.S. Ramesh and M.H. Lean. Accurate integration of singular kernels in boundary integral formulations for Helmholtz-equation. *International Journal for Numerical Methods in Engineering*, 31(6):1055–1068, 1991.

[75] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60:187–207, 1985.

[76] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *Journal of Computational Physics*, 86:414–439, 1990.

[77] H. Schulz, C. Schwab, and W.L. Wendland. An extraction technique for boundary element methods. In W. Hackbusch and G. Wittum, editors, *Boundary Elements: Implementation and Analysis of Advanced Algorithms. Proceedings of the Twelfth GAMM-Seminar, Kiel*, pages 219–231, January 1996.

[78] C. Schwab and W.L. Wendland. On the extraction technique in boundary integral equations. Technical Report 96-3, Mathematics Institute A, Stuttgart University, 1996.

[79] I.H. Sloan. Polynomial interpolation and hyperinterpolation over general regions. *Journal of Approximation Theory*, 83(2):238–254, 1995.

[80] G. Strang. *Introduction to Applied Mathematics.* Wellesley Cambridge Press, 1986.

[81] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* Wadsworth and Brooks / Cole Mathematics Series. Wadsworth, Pacific Grove, California, 1989.

[82] G. Vainikko. *Multidimensional Weakly Singular Integral Equations.* Springer-Verlag, Berlin, 1993.

[83] S. Wang, I.H. Sloan, and D.W. Kelly. Computable error bounds for pointwise derivatives of a Neumann problem. *IMA Journal of Numerical Analysis*, 18:251–271, 1998.